From Python to Base R: Institutional & Field-of-Study Analytics

Will, Leo, James

October 14, 2025

Goals

- Translate Python to base R, no tidyverse or dplyr.
- Keep the **same analysis intent**: cleaning, profiling, mapping codes, ROI, regions, locale, and field-of-study.
- Provide **side-by-side explanations**

Conventions Used

- Use `read.csv()` and `read.delim()` to import data.
- Use **named vectors** for code maps.
- Use `as.numeric()`, `is.na()`, `complete.cases()` for cleaning.
- Use `aggregate()`, `tapply()`, `by()` for grouped summaries.
- Use **base graphics**: `plot()`, `points()`, `barplot()`, `boxplot()`, `legend()`.

---

1) Institutional Columns to Load

```
data_dir <- "College_Scorecard_Raw_Data_05192025"

columns_to_load_institution <- c(
  "UNITID","INSTNM","CONTROL","PREDDEG","HIGHDEG",
  "REGION","LOCALE","UGDS",
  "COSTT4_A","COSTT4_P",
  "TUITIONFEE_IN","TUITIONFEE_OUT","TUITIONFEE_PROG",
  "MD_EARN_WNE_P10",
  "NPT4_PUB","NPT4_PRIV","NPT4_PROG",
  "C150_4","C150_L4","C150_4_POOLED","C200_L4","C200_4"
)
print(data_dir)
```

```
## [1] "College_Scorecard_Raw_Data_05192025"
```

Import (base R):

```
# Update the path file as needed
inst_path <- file.path(data_dir, "Most-Recent-Cohorts-Institution.csv")

df <- read.csv(inst_path, stringsAsFactors = FALSE)
# Keep only columns that exist
df <- df[ , intersect(columns_to_load_institution, names(df))]
```

```r
cat("Original shape:", nrow(df), "rows x", ncol(df), "cols\n")
```

```
## Original shape: 6429 rows x 22 cols
```

```r
head(df, 3)
```

```
##   UNITID                               INSTNM CONTROL PREDDEG HIGHDEG REGION
## 1 100654             Alabama A & M University       1       3       4      5
## 2 100663 University of Alabama at Birmingham       1       3       4      5
## 3 100690                   Amridge University       2       3       4      5
##   LOCALE  UGDS COSTT4_A COSTT4_P TUITIONFEE_IN TUITIONFEE_OUT TUITIONFEE_PROG
## 1     12  5726    23751       NA         10024          18634              NA
## 2     12 12118    27826       NA          8832          21864              NA
## 3     12   226       NA       NA            NA             NA              NA
##   MD_EARN_WNE_P10 NPT4_PUB NPT4_PRIV NPT4_PROG C150_4 C150_L4 C150_4_POOLED
## 1           40628    14559        NA        NA 0.2874      NA        0.2772
## 2           54501    17727        NA        NA 0.6260      NA        0.6345
## 3           37621       NA        NA        NA 0.4000      NA        0.4000
##   C200_L4 C200_4
## 1      NA 0.2962
## 2      NA 0.6490
## 3      NA 0.6667
```

---

2) ROI Columns to Numeric & Missing Profile

```r
roi_cols <- c("COSTT4_A","MD_EARN_WNE_P10")
for (col in roi_cols) {
  # Coerce; non-numeric like 'PS' becomes NA
  df[[col]] <- suppressWarnings(as.numeric(df[[col]]))
}

cat("Missing values in ROI columns:\n")
```

```
## Missing values in ROI columns:
```

```r
print(colSums(is.na(df[roi_cols])))
```

```
##        COSTT4_A MD_EARN_WNE_P10
##            3182            1149
```

---

3) Profile Non-reporting (Missing COSTT4_A)

```r
df_missing_cost <- df[ is.na(df$COSTT4_A), ]

# CONTROL map (1/2/3)
control_map <- c("1"="Public","2"="Private Nonprofit","3"="Private For-Profit")
df_missing_cost$CONTROL_NAME <- control_map[ as.character(df_missing_cost$CONTROL) ]

cat(nrow(df_missing_cost), "institutions missing COSTT4_A\n")
```

```
## 3182 institutions missing COSTT4_A
```

```r
ctrl_tb <- table(df_missing_cost$CONTROL_NAME)
ctrl_pct <- round(100 * prop.table(ctrl_tb), 2)
print(ctrl_tb); print(ctrl_pct)
```

```
##
## Private For-Profit  Private Nonprofit             Public
##               2060                642                480
```

```
##
## Private For-Profit  Private Nonprofit             Public
##              64.74              20.18              15.08
```

```r
# PREDDEG map
preddeg_map <- c(
  "0"="Not Classified",
  "1"="Predominantly Certificate",
  "2"="Predominantly Associate's",
  "3"="Predominantly Bachelor's",
  "4"="Exclusively Graduate"
)
df_missing_cost$PREDDEG_NAME <- preddeg_map[ as.character(df_missing_cost$PREDDEG) ]
table(df_missing_cost$PREDDEG_NAME)
```

```
##
##      Exclusively Graduate            Not Classified Predominantly Associate's
##                       280                       507                        91
##  Predominantly Bachelor's Predominantly Certificate
##                       186                      2118
```

Of the 3,182 institutions that are missing cost data, nearly two-thirds (65%) are Private For-Profit.

These are mostly certificate-granting institutions, not traditional 4-year colleges.

---

4) Cleaned Dataset (Drop NA in ROI)

```r
df_cleaned <- df[ complete.cases(df[roi_cols]), ]
cat("After drop-NA (ROI):", nrow(df_cleaned), "rows\n")
```

```
## After drop-NA (ROI): 3075 rows
```

```r
summary(df_cleaned[roi_cols])
```

```
##     COSTT4_A     MD_EARN_WNE_P10
##  Min.   : 4274   Min.   : 11998
##  1st Qu.:15710   1st Qu.: 37844
##  Median :24702   Median : 45388
##  Mean   :30595   Mean   : 48518
##  3rd Qu.:41526   3rd Qu.: 56316
##  Max.   :87804   Max.   :143372
```

After removing all rows with missing cost or earnings data, the final dataset is 3,075 institutions.

---

5) Field-of-Study Import & Cleaning

```r
# Path placeholders; update as needed if available
fos_path <- file.path(data_dir, "Most-Recent-Cohorts-Field-of-Study.csv")

if (file.exists(fos_path)) {
  df_field <- read.csv(fos_path, stringsAsFactors = FALSE)

  # Keep relevant columns if present
  columns_to_load_field <- c(
    "EARN_GT_THRESHOLD_1YR","EARN_GT_THRESHOLD_5YR",
    "UNITID","CIPCODE","CREDLEV","CONTROL",
    "EARN_MDN_1YR","EARN_MDN_4YR"
  )
  df_field <- df_field[ , intersect(columns_to_load_field, names(df_field)) ]

  # Coerce and cap > 100 to NA
  make_num <- function(x) suppressWarnings(as.numeric(x))
  if ("EARN_GT_THRESHOLD_1YR" %in% names(df_field)) {
    df_field$EARN_GT_THRESHOLD_1YR <- make_num(df_field$EARN_GT_THRESHOLD_1YR)
    df_field$EARN_GT_THRESHOLD_1YR[df_field$EARN_GT_THRESHOLD_1YR > 100] <- NA
  }
  if ("EARN_GT_THRESHOLD_5YR" %in% names(df_field)) {
    df_field$EARN_GT_THRESHOLD_5YR <- make_num(df_field$EARN_GT_THRESHOLD_5YR)
    df_field$EARN_GT_THRESHOLD_5YR[df_field$EARN_GT_THRESHOLD_5YR > 100] <- NA
  }

  summary(df_field)
}
```

```
##  EARN_GT_THRESHOLD_1YR EARN_GT_THRESHOLD_5YR     UNITID          CIPCODE
##  Min.   : 16.0         Min.   : 16.00        Min.   :100654   Min.   : 100
##  1st Qu.: 22.0         1st Qu.: 22.00        1st Qu.:149231   1st Qu.:1433
##  Median : 32.0         Median : 32.00        Median :187532   Median :4001
##  Mean   : 38.8         Mean   : 38.77        Mean   :202261   Mean   :3321
##  3rd Qu.: 51.0         3rd Qu.: 50.00        3rd Qu.:220978   3rd Qu.:5107
##  Max.   :100.0         Max.   :100.00        Max.   :497338   Max.   :6127
##  NA's   :196450        NA's   :193211        NA's   :10109
##     CREDLEV        CONTROL           EARN_MDN_1YR       EARN_MDN_4YR
##  Min.   :1.00   Length:229188      Length:229188      Length:229188
##  1st Qu.:2.00   Class :character   Class :character   Class :character
##  Median :3.00   Mode  :character   Mode  :character   Mode  :character
##  Mean   :3.27
##  3rd Qu.:5.00
##  Max.   :8.00
##
```

---

6) Predominant Degree Share (Counts & Percent)

```r
df_cleaned$PREDDEG_NAME <- preddeg_map[ as.character(df_cleaned$PREDDEG) ]
roi_counts <- table(df_cleaned$PREDDEG_NAME)
roi_percent <- round(100 * prop.table(roi_counts), 2)
```

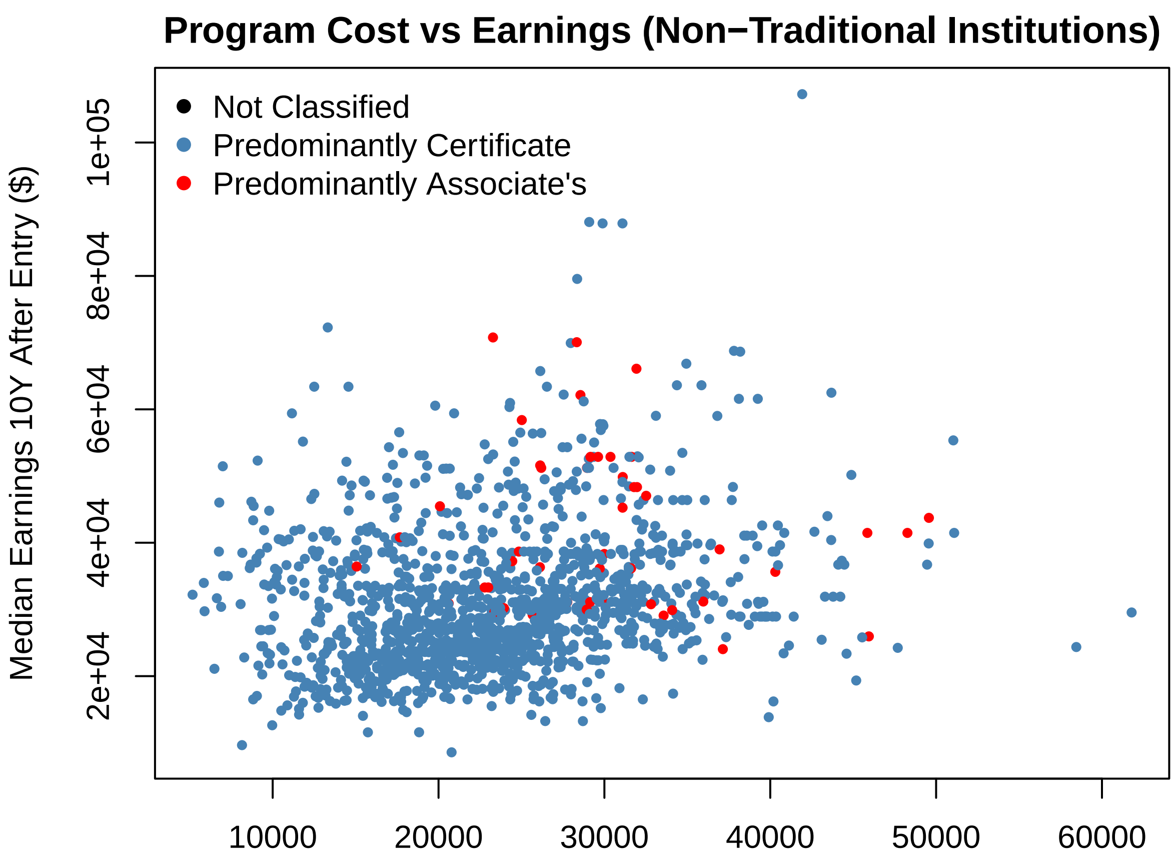
```r
with_roi <- data.frame(Count = as.vector(roi_counts),
                       Percentage = as.vector(roi_percent),
                       row.names = names(roi_counts))
print(with_roi)
```

```
##                              Count Percentage
## Predominantly Associate's      855      27.80
## Predominantly Bachelor's      1707      55.51
## Predominantly Certificate      513      16.68
```

---

7) Non-Traditional Institutions: Program Cost vs Earnings

```r
# Targets: Not Classified, Certificate, Associate's => codes 0,1,2
target_preddegs <- c(0,1,2)
# Ensure PREDDEG is numeric
df$PREDDEG <- suppressWarnings(as.numeric(df$PREDDEG))

df_missing_academic_cost <- df[ is.na(df$COSTT4_A), ]
df_target <- df_missing_academic_cost[ df_missing_academic_cost$PREDDEG %in% target_preddegs, ]

# Keep rows with both program cost & earnings
keep <- !is.na(df_target$COSTT4_P) & !is.na(df_target$MD_EARN_WNE_P10)
df_plot_ready <- df_target[ keep, ]

df_plot_ready$PREDDEG_NAME <- preddeg_map[ as.character(df_plot_ready$PREDDEG) ]

par(mar = c(2, 4.1, 2, 2))
# Base R scatter, color by PREDDEG_NAME
cols <- c("Not Classified"="black","Predominantly Certificate"="steelblue","Predominantly Associate's"=
plot(df_plot_ready$COSTT4_P, df_plot_ready$MD_EARN_WNE_P10,
     xlab="Average Annual Program Cost ($)", ylab="Median Earnings 10Y After Entry ($)",
     main="Program Cost vs Earnings (Non-Traditional Institutions)",
     col = cols[df_plot_ready$PREDDEG_NAME], pch=16, cex=0.7);
legend("topleft", legend=names(cols), col=cols, pch=16, bty="n")
```



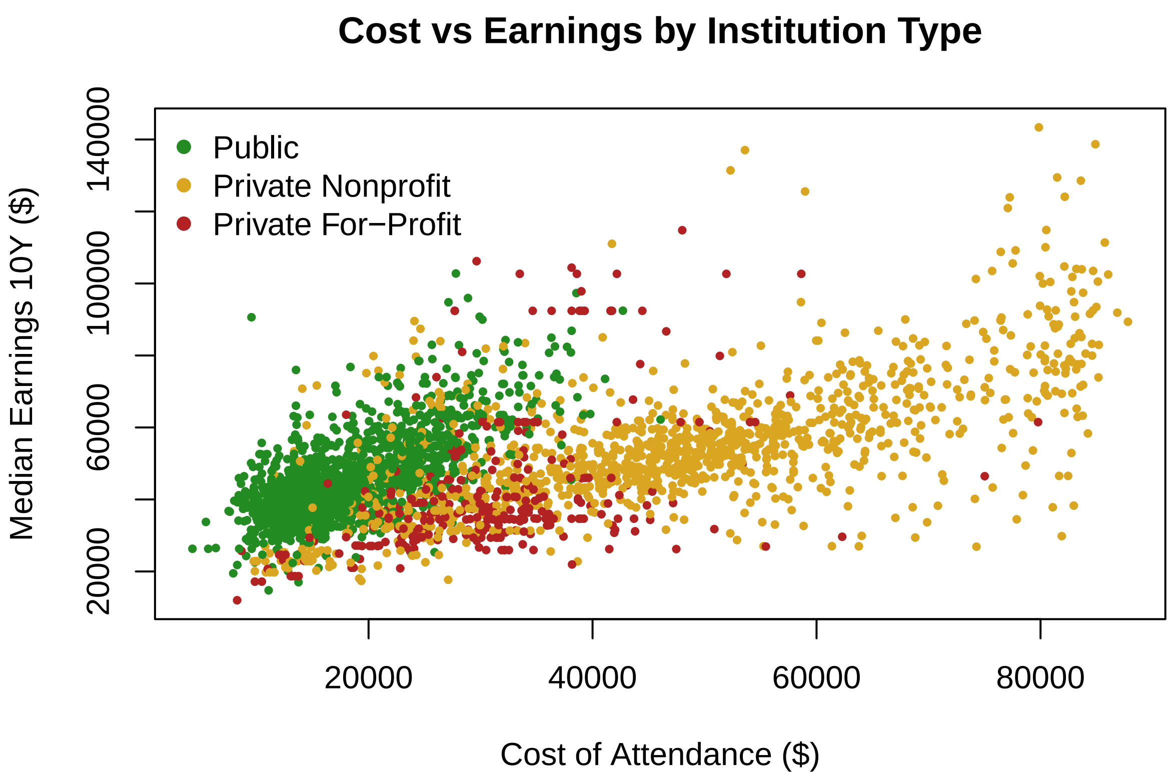
---

8) Institution Type & Scatter (Correct CONTROL Mapping)

```
df_cleaned$CONTROL_NAME <- control_map[ as.character(df_cleaned$CONTROL) ]
inst_counts <- table(df_cleaned$CONTROL_NAME)
inst_pct <- round(100 * prop.table(inst_counts), 2)
data.frame(Count=as.vector(inst_counts), Percentage=as.vector(inst_pct),
           row.names = names(inst_counts))
```

```
##                      Count Percentage
## Private For-Profit     327      10.63
## Private Nonprofit     1190      38.70
## Public                1558      50.67
```

```
# Cost vs Earnings colored by institution type
cols2 <- c("Public"="forestgreen","Private Nonprofit"="goldenrod","Private For-Profit"="firebrick")
plot(df_cleaned$COSTT4_A, df_cleaned$MD_EARN_WNE_P10,
     xlab="Cost of Attendance ($)", ylab="Median Earnings 10Y ($)",
     main="Cost vs Earnings by Institution Type",
     col=cols2[df_cleaned$CONTROL_NAME], pch=16, cex=0.6);
legend("topleft", legend=names(cols2), col=cols2, pch=16, bty="n")
```



---

9) Net Price, ROI, and Top-10

```
# If public net price missing, use private net price
df$NPT4_PUB <- suppressWarnings(as.numeric(df$NPT4_PUB))
df$NPT4_PRIV <- suppressWarnings(as.numeric(df$NPT4_PRIV))

df$NET_PRICE <- ifelse(!is.na(df$NPT4_PUB), df$NPT4_PUB, df$NPT4_PRIV)
df_np <- df[ !is.na(df$NET_PRICE) & !is.na(df$MD_EARN_WNE_P10), ]
cat("Found", nrow(df_np), "institutions with Net Price & Earnings\n")
```

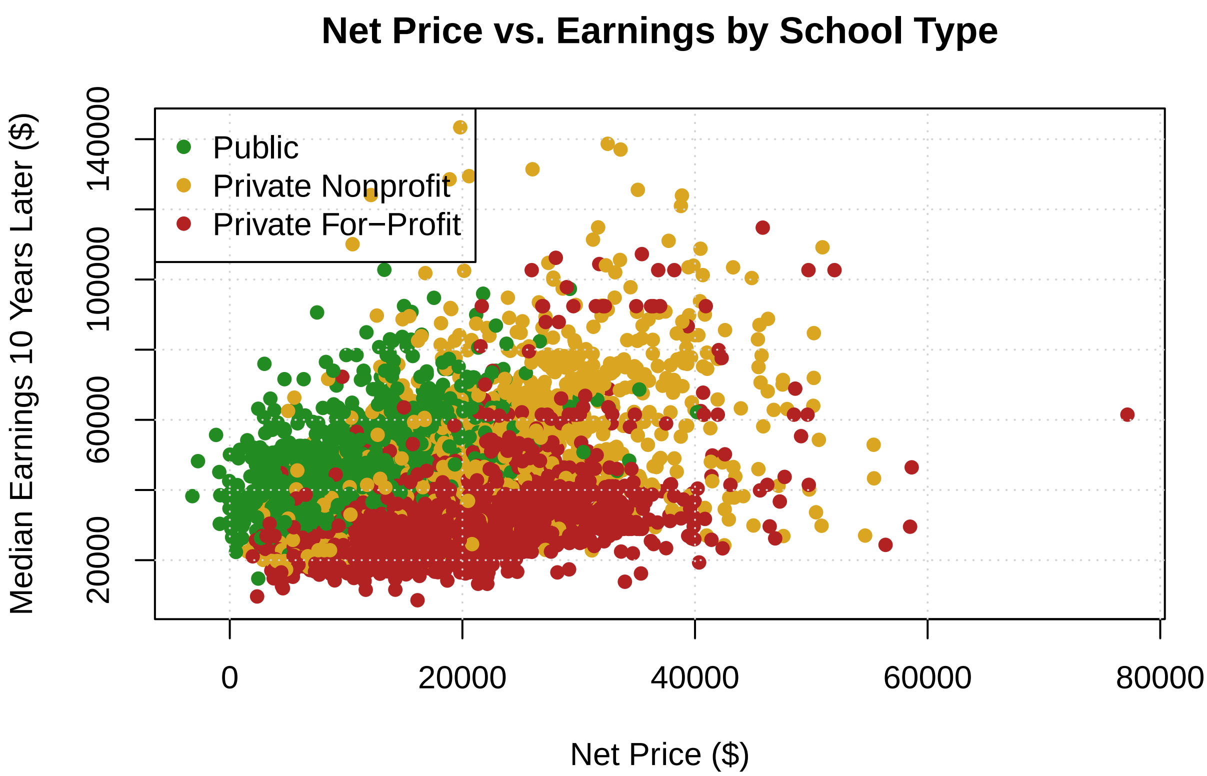
```
## Found 4541 institutions with Net Price & Earnings
```

```
df_np$SCHOOL_TYPE <- control_map[ as.character(df_np$CONTROL) ]
df_np$ROI_RATIO <- df_np$MD_EARN_WNE_P10 / df_np$NET_PRICE

# Top-10 overall for earnings > 80k, highest ROI
top_overall <- df_np[ df_np$MD_EARN_WNE_P10 > 80000, c("INSTNM","ROI_RATIO","MD_EARN_WNE_P10","COSTT4_A
top_overall <- top_overall[ order(-top_overall$ROI_RATIO), ]
head(top_overall, 10)
```

```
##                                        INSTNM ROI_RATIO MD_EARN_WNE_P10
## 2203       United States Merchant Marine Academy 12.071676           90610
## 1902                        Princeton University 10.427854          110066
## 3580                         Stanford University 10.224127          124080
## 743   Georgia Institute of Technology-Main Campus  7.733614          102772
## 1413       Massachusetts Institute of Technology  7.236259          143372
## 212           University of California-San Diego  7.229191           84943
## 3157                             Rice University  7.097943           89718
```

```
## 190           California Institute of Technology  6.801714          128566
## 4287      Franklin W Olin College of Engineering  6.291859          129455
## 209              University of California-Irvine  6.287773           80735
##      COSTT4_A NET_PRICE
## 2203     9547      7506
## 1902    80440     10555
## 3580    82162     12136
## 743     27797     13289
## 1413    79850     19813
## 212     36325     11750
## 3157    74110     12640
## 190     83598     18902
## 4287    81486     20575
## 209     36121     12840
```

```r
df_np$school_type_name <- ifelse(df_np$CONTROL == 1, "Public",
                                 ifelse(df_np$CONTROL == 2, "Private Nonprofit",
                                        "Private For-Profit"))
df_np$plot_color <- ifelse(df_np$school_type_name == "Public", "forestgreen",
                           ifelse(df_np$school_type_name == "Private Nonprofit", "goldenrod",
                                  "firebrick"))
plot(
  x = df_np$NET_PRICE,
  y = df_np$MD_EARN_WNE_P10,
  main = "Net Price vs. Earnings by School Type",
  xlab = "Net Price ($)",
  ylab = "Median Earnings 10 Years Later ($)",

  col = df_np$plot_color,
  pch = 16
)


grid()

legend(
  "topleft",
  legend = c("Public", "Private Nonprofit", "Private For-Profit"),
  col = c("forestgreen", "goldenrod", "firebrick"),
  pch = 16
)
```



---

10) Top-10 by Institution Type

```
top_by_type <- function(ctrl_code) {
  tmp <- df_np[ df_np$CONTROL == ctrl_code & df_np$MD_EARN_WNE_P10 > 80000,
               c("INSTNM","ROI_RATIO","MD_EARN_WNE_P10","COSTT4_A","NET_PRICE") ]
  tmp <- tmp[ order(-tmp$ROI_RATIO), ]
  head(tmp, 10)
}
top_public  <- top_by_type(1)
top_priv_np <- top_by_type(2)
top_priv_fp <- top_by_type(3)

top_public; top_priv_np; top_priv_fp
```

```
##                                                    INSTNM ROI_RATIO
## 2203                  United States Merchant Marine Academy 12.071676
## 743             Georgia Institute of Technology-Main Campus  7.733614
## 212                      University of California-San Diego  7.229191
## 209                         University of California-Irvine  6.287773
## 207                       University of California-Berkeley  6.171707
## 1723            Missouri University of Science and Technology  6.023161
## 210                     University of California-Los Angeles  5.888175
## 192  California Polytechnic State University-San Luis Obispo  5.809524
## 1519                        University of Michigan-Ann Arbor  5.639698
## 221               California State University Maritime Academy  5.399259
##      MD_EARN_WNE_P10 COSTT4_A NET_PRICE
## 2203           90610     9547      7506
```

```
## 743          102772    27797     13289
## 212           84943    36325     11750
## 209           80735    36121     12840
## 207           92446    42708     14979
## 1723          82957    25653     13773
## 210           82511    36643     14013
## 192           90768    29918     15624
## 1519          83648    33345     14832
## 221           94784    27138     17555
```

```
##                                     INSTNM ROI_RATIO MD_EARN_WNE_P10 COSTT4_A
## 1902                  Princeton University 10.427854          110066    80440
## 3580                   Stanford University 10.224127          124080    82162
## 1413  Massachusetts Institute of Technology  7.236259          143372    79850
## 3157                       Rice University  7.097943           89718    74110
## 190     California Institute of Technology  6.801714          128566    83598
## 4287 Franklin W Olin College of Engineering  6.291859          129455    81486
## 1397                    Harvard University  6.054769          101817    82842
## 1459                      Williams College  5.969903           88665    81164
## 4271      Yeshiva Shaarei Torah of Rockland  5.798239           89548    24095
## 4589                 Yeshivas Be'er Yitzchok  5.099759           82560    32040
##      NET_PRICE
## 1902     10555
## 3580     12136
## 1413     19813
## 3157     12640
## 190      18902
## 4287     20575
## 1397     16816
## 1459     14852
## 4271     15444
## 4589     16189
```

```
##                                    INSTNM ROI_RATIO MD_EARN_WNE_P10 COSTT4_A
## 5475 Chamberlain University-North Carolina  4.265371           92405    27697
## 5066          West Coast University-Dallas  3.955465          102672    33502
## 768              Miami Ad School-Atlanta  3.788242          106192    29647
## 4443              United States University  3.756552           80980    28358
## 5358        Chamberlain University-Michigan  3.438965           92405    27697
## 4626        Chamberlain University-Illinois  3.430921           92405    36346
## 4401    Neumont College of Computer Science  3.375789           97827    39004
## 1991     St Paul's School of Nursing-Queens  3.286627          104403    38134
## 5074                         Unitek College  3.235768           87877       NA
## 4743         Chamberlain University-Florida  3.128237           92405    34657
##      NET_PRICE
## 5475     21664
## 5066     25957
## 768      28032
## 4443     21557
## 5358     26870
## 4626     26933
## 4401     28979
## 1991     31766
## 5074     27158
## 4743     29539
```

---

11) Regions

```r
region_map <- c(
  "0"="U.S. Service Schools","1"="New England","2"="Mid East","3"="Great Lakes",
  "4"="Plains","5"="Southeast","6"="Southwest","7"="Rocky Mountains","8"="Far West","9"="Outlying Areas
)
df_cleaned$NET_PRICE <- ifelse(!is.na(df_cleaned$NPT4_PUB), df_cleaned$NPT4_PUB, df_cleaned$NPT4_PRIV)
df_cleaned$REGION_NAME <- region_map[ as.character(df_cleaned$REGION) ]
table(df_cleaned$REGION_NAME)
```

```
##
##             Far West          Great Lakes             Mid East
##                  362                  454                  500
##          New England       Outlying Areas               Plains
##                  183                   85                  308
##      Rocky Mountains            Southeast            Southwest
##                  101                  807                  274
## U.S. Service Schools
##                    1
```

```r
# Region means and diff
region_means <- aggregate(cbind(MD_EARN_WNE_P10, NET_PRICE) ~ REGION_NAME, df_cleaned, mean, na.rm=TRUE
region_means$DIFF_ABS <- region_means$MD_EARN_WNE_P10 - region_means$NET_PRICE
region_means[ order(-region_means$DIFF_ABS), ]
```

```
##             REGION_NAME MD_EARN_WNE_P10 NET_PRICE DIFF_ABS
## 10 U.S. Service Schools        90610.00  7506.000 83104.00
## 1              Far West        52716.48 16389.163 36327.32
## 4           New England        59599.73 23583.492 36016.24
## 3              Mid East        55432.55 20418.926 35013.62
## 2           Great Lakes        48886.79 16634.672 32252.12
## 6                Plains        48914.53 16862.981 32051.55
## 7       Rocky Mountains        47153.48 15775.545 31377.93
## 9             Southwest        45092.89 14404.310 30688.58
## 8             Southeast        43182.31 15635.817 27546.49
## 5        Outlying Areas        25522.11  7214.165 18307.94
```

---

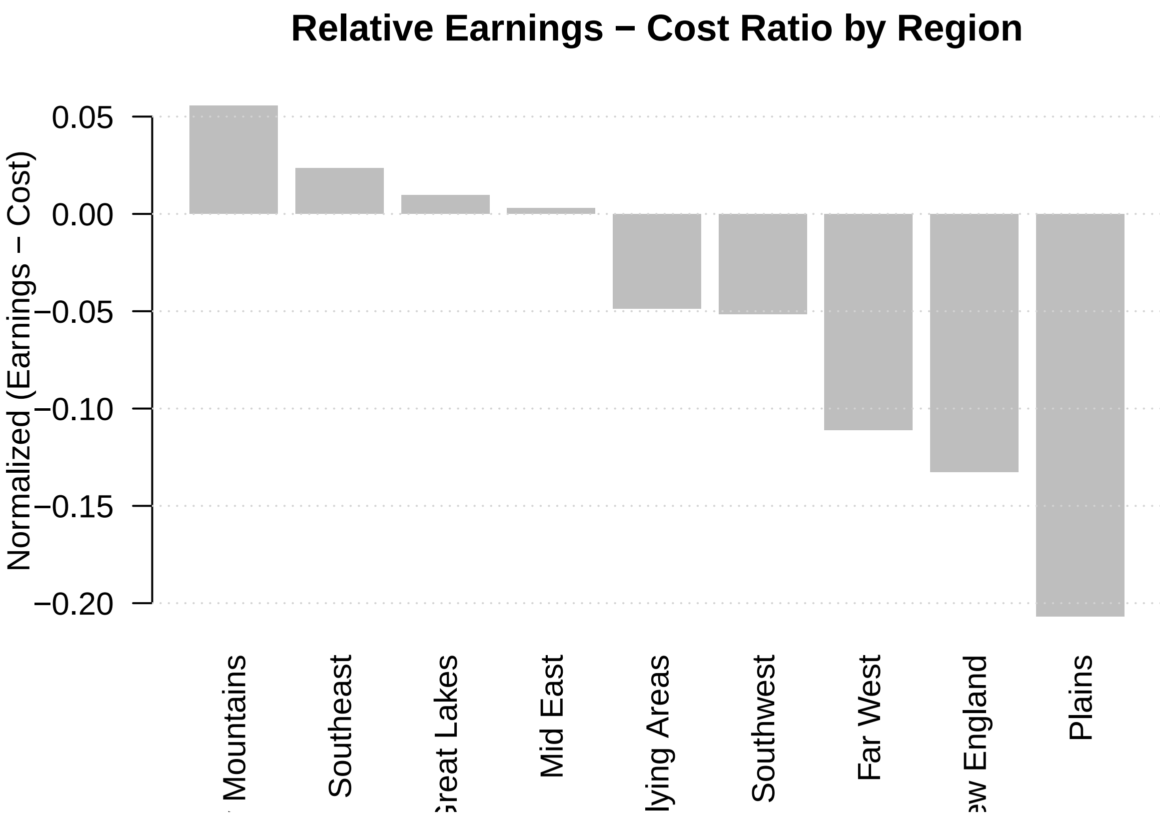
12) Region-Level Normalization & Trend

```r
# Normalize within region: (x - min) / (max - min)
norm_in_group <- function(x) (x - min(x, na.rm=TRUE)) / (max(x, na.rm=TRUE) - min(x, na.rm=TRUE))

df_cleaned$earn_norm <- ave(df_cleaned$MD_EARN_WNE_P10, df_cleaned$REGION_NAME, FUN=norm_in_group)
df_cleaned$net_norm  <- ave(df_cleaned$NET_PRICE,       df_cleaned$REGION_NAME, FUN=norm_in_group)
df_cleaned$net_diff_ratio <- df_cleaned$earn_norm - df_cleaned$net_norm

region_trends <- aggregate(net_diff_ratio ~ REGION_NAME, df_cleaned, mean, na.rm=TRUE)
region_trends <- region_trends[ order(-region_trends$net_diff_ratio), ]
region_trends
```

```
##         REGION_NAME net_diff_ratio
```

```
## 7 Rocky Mountains    0.055545083
## 8       Southeast    0.023625876
## 2     Great Lakes    0.009672379
## 3        Mid East    0.002902008
## 5  Outlying Areas   -0.048630360
## 9       Southwest   -0.051484706
## 1        Far West   -0.110921508
## 4     New England   -0.132540574
## 6          Plains   -0.206690954
```

```r
# Barplot
barplot(height = region_trends$net_diff_ratio, names.arg = region_trends$REGION_NAME,
        las=2, main="Relative Earnings - Cost Ratio by Region",
        ylab="Normalized (Earnings - Cost)", border=NA)
grid(nx=NA, ny=NULL)
```



13) Locale Simplification & Plots

```r
simplify_locale <- function(locale_code) {
  if (is.na(locale_code)) return("Unknown")
  if (locale_code >= 11 && locale_code <= 13) return("City")
  if (locale_code >= 21 && locale_code <= 23) return("Suburb")
  if (locale_code >= 31 && locale_code <= 33) return("Town")
  if (locale_code >= 41 && locale_code <= 43) return("Rural")
  "Unknown"
}
```

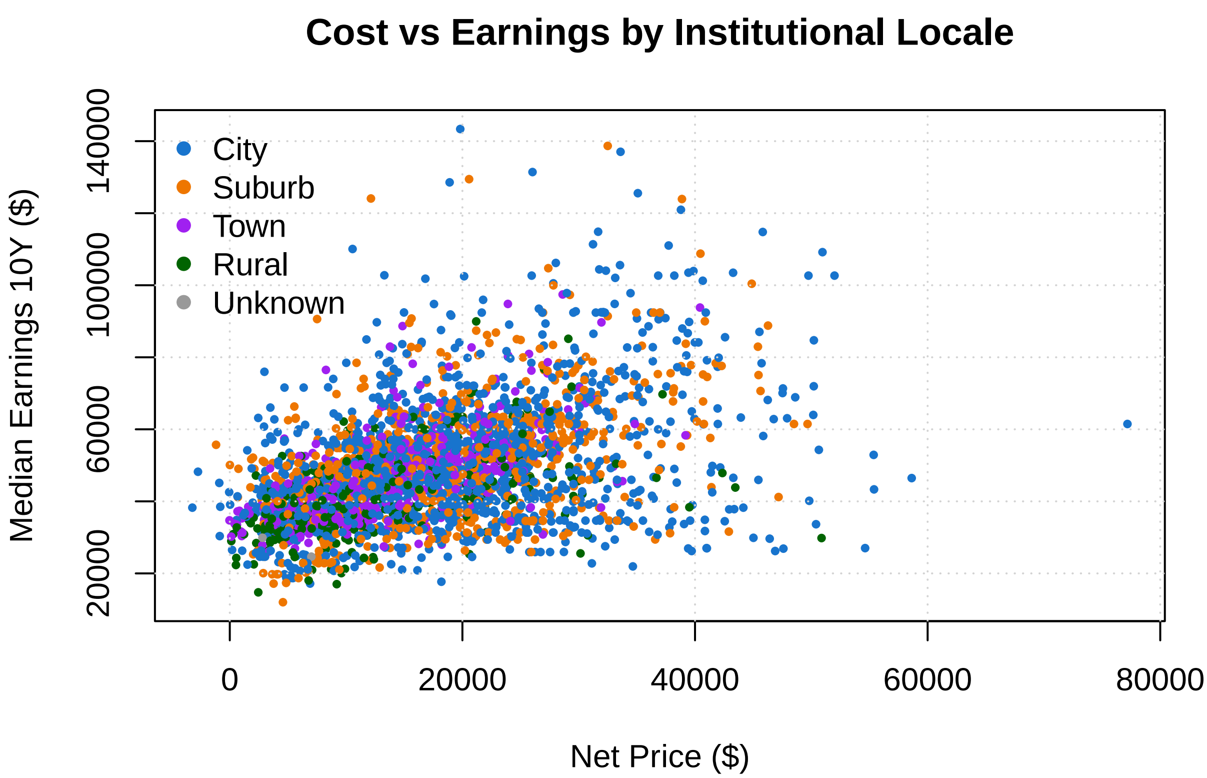
```r
df_cleaned$LOCALE_TYPE <- sapply(df_cleaned$LOCALE, simplify_locale)
table(df_cleaned$LOCALE_TYPE)
```

```
##
##    City   Rural  Suburb    Town Unknown
##    1385     385     731     572       2
```

```r
# Locale summary means
loc_means <- aggregate(cbind(NET_PRICE, MD_EARN_WNE_P10) ~ LOCALE_TYPE, df_cleaned, mean, na.rm=TRUE)
loc_means$DIFF <- loc_means$MD_EARN_WNE_P10 - loc_means$NET_PRICE
loc_means
```

```
##   LOCALE_TYPE NET_PRICE MD_EARN_WNE_P10     DIFF
## 1        City  18623.07        50458.80 31835.73
## 2       Rural  12598.77        41174.21 28575.44
## 3      Suburb  18160.30        50527.69 32367.39
## 4        Town  14081.31        46267.38 32186.07
## 5     Unknown   4911.00        27274.50 22363.50
```

```r
# Scatter by locale
loc_cols <- c("City"="dodgerblue3","Suburb"="darkorange2","Town"="purple","Rural"="darkgreen","Unknown"=
plot(df_cleaned$NET_PRICE, df_cleaned$MD_EARN_WNE_P10,
     xlab="Net Price ($)", ylab="Median Earnings 10Y ($)",
     main="Cost vs Earnings by Institutional Locale",
     col=loc_cols[df_cleaned$LOCALE_TYPE], pch=16, cex=0.6)
grid(); legend("topleft", legend=names(loc_cols), col=loc_cols, pch=16, bty="n")
```



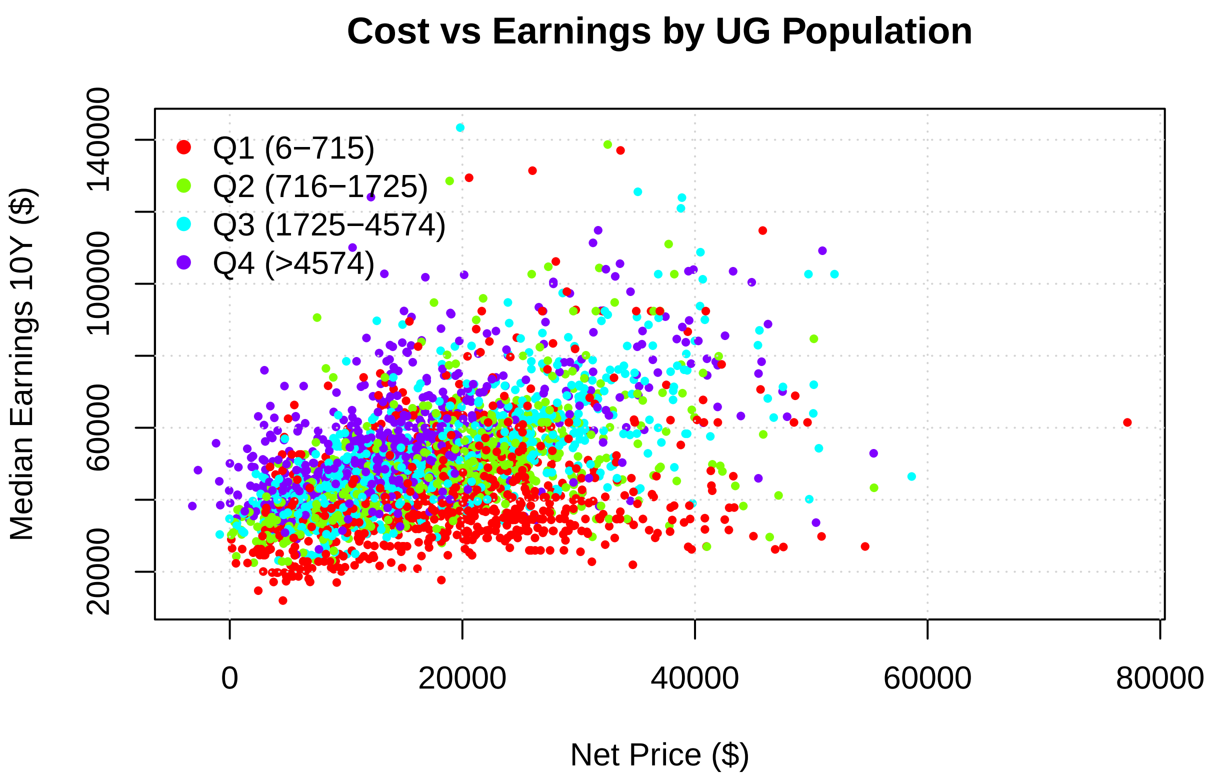
14) UGDS Quartiles & Size Plot

```
summary(df_cleaned$UGDS)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       6     716    1725    4319    4574  156755
```

```
# Use exact cut points as in Python example
breaks <- c(0, 716, 1725, 4574, Inf)
labels <- c("Q1 (6-715)","Q2 (716-1725)","Q3 (1725-4574)","Q4 (>4574)")
df_cleaned$SCHOOL_SIZE <- cut(df_cleaned$UGDS, breaks=breaks, labels=labels, right=FALSE)

sz_cols <- setNames(rainbow(length(labels)), labels)
```

```
plot(df_cleaned$NET_PRICE, df_cleaned$MD_EARN_WNE_P10,
     xlab="Net Price ($)", ylab="Median Earnings 10Y ($)",
     main="Cost vs Earnings by UG Population",
     col=sz_cols[df_cleaned$SCHOOL_SIZE], pch=16, cex=0.6)
grid(); legend("topleft", legend=labels, col=sz_cols, pch=16, bty="n")
```

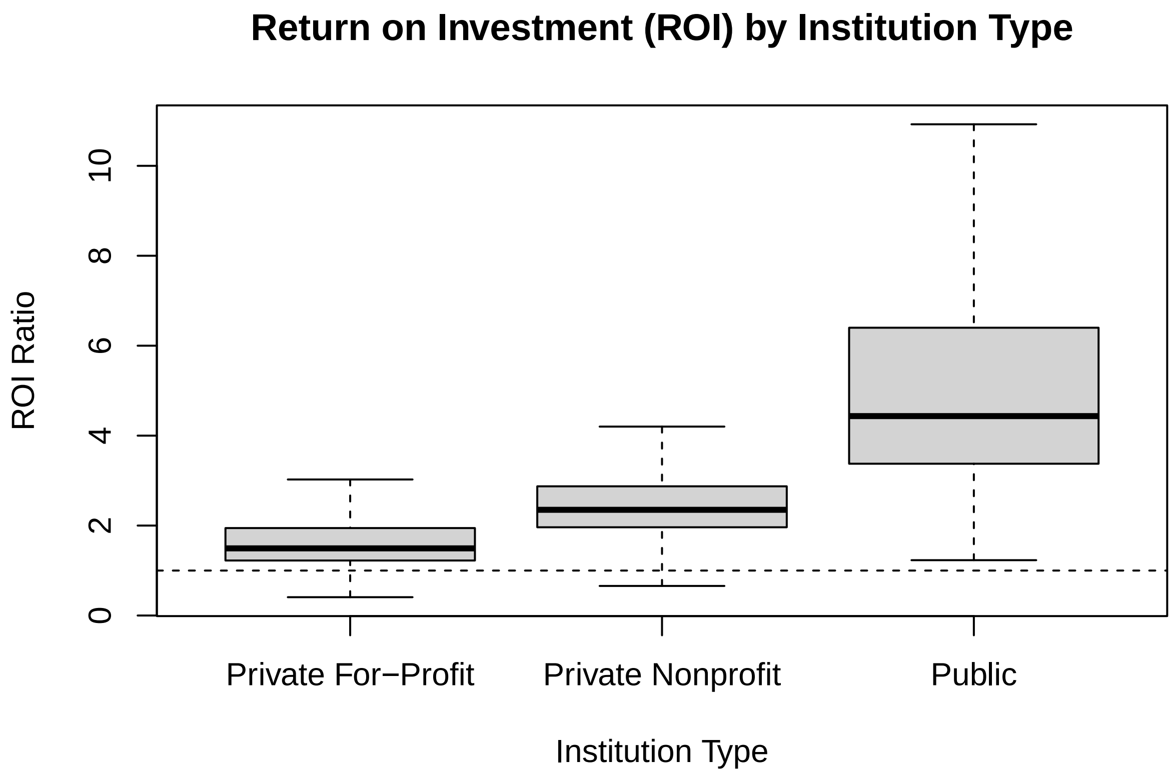


---

15) ROI by Institution Type (Boxplot) & Medians

```
# Ensure ROI present (df_np has NET_PRICE & MD_EARN_WNE_P10)
# Reuse df_np from earlier
if (!exists("df_np")) {
  df$NPT4_PUB <- suppressWarnings(as.numeric(df$NPT4_PUB))
  df$NPT4_PRIV <- suppressWarnings(as.numeric(df$NPT4_PRIV))
  df$NET_PRICE <- ifelse(!is.na(df$NPT4_PUB), df$NPT4_PUB, df$NPT4_PRIV)
```

```r
  df_np <- df[ !is.na(df$NET_PRICE) & !is.na(df$MD_EARN_WNE_P10), ]
  df_np$SCHOOL_TYPE <- control_map[ as.character(df_np$CONTROL) ]
  df_np$ROI_RATIO <- df_np$MD_EARN_WNE_P10 / df_np$NET_PRICE
}

boxplot(ROI_RATIO ~ SCHOOL_TYPE, data=df_np,
        main="Return on Investment (ROI) by Institution Type",
        ylab="ROI Ratio", xlab="Institution Type",
        outline=FALSE)
abline(h=1, lty=2)
```




```r
tapply(df_np$ROI_RATIO, df_np$SCHOOL_TYPE, median, na.rm=TRUE)
```

```
## Private For-Profit  Private Nonprofit             Public
##           1.493312           2.350606           4.434614
```

---

17) Completion Rate Cleaning & Overview

```r
df_with_cost <- df[!is.na(df$COSTT4_A), ]

completion_rates <- suppressWarnings(as.numeric(df_with_cost$C150_4_POOLED))

compl_clean <- completion_rates[!is.na(completion_rates)]

cat("Total institutions with cost data:", nrow(df_with_cost), "\n\n")
```

```
## Total institutions with cost data: 3247
```

```r
cat("Of those, institutions with valid completion data:", length(compl_clean), "\n\n")
```

```
## Of those, institutions with valid completion data: 2151
```

```r
cat("Missing rate within this subset (%):", round(100 * (1 - length(compl_clean) / nrow(df_with_cost)),
```

```
## Missing rate within this subset (%): 33.75
```

```r
summary(compl_clean)
```

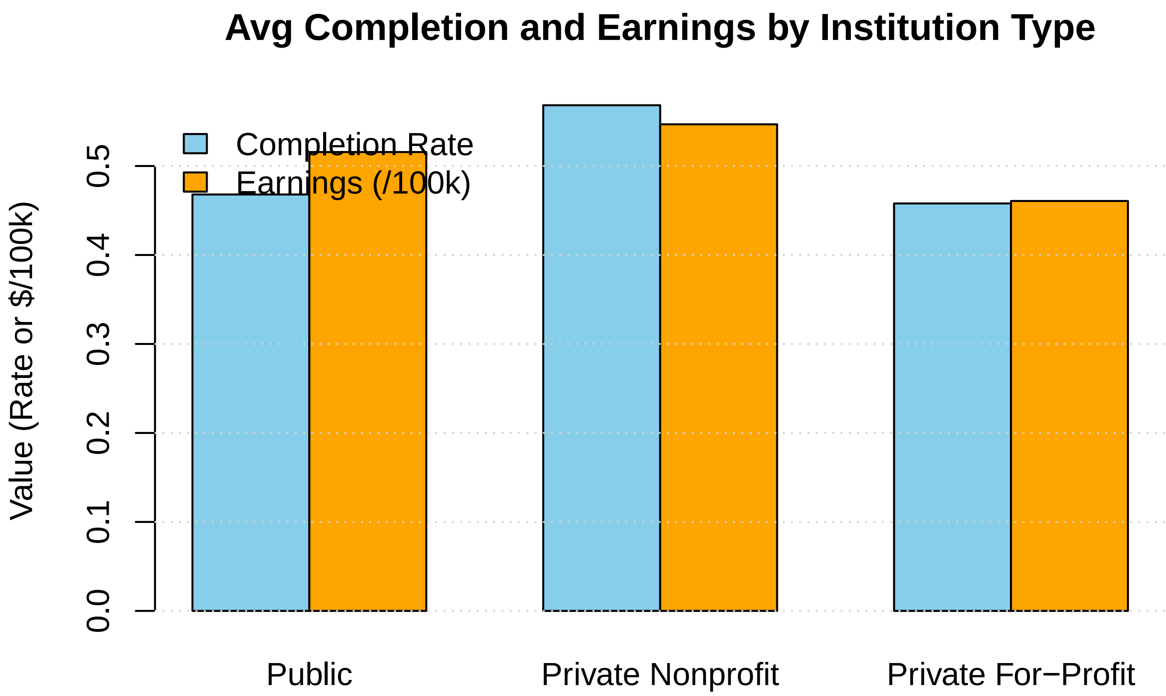
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.0000  0.3735  0.5213  0.5185  0.6622  1.0000
```

---

19) Completion & Earnings by Institution Type

```r
needed <- c("CONTROL","C150_4_POOLED","MD_EARN_WNE_P10")
if (all(needed %in% names(df))) {

  # Aggregate directly from the original 'df' data frame
  summary_by_type <- aggregate(
    cbind(C150_4_POOLED, MD_EARN_WNE_P10) ~ CONTROL,
    data = df,
    FUN = mean,
    na.action = na.omit
  )

  # 2. Map the control codes to names for labeling.
  c(1, 2, 3)
  as.character( c(1, 2, 3) )
  c("Public", "Private Nonprofit", "Private For-Profit")
  summary_by_type$CONTROL_NAME <- c("Public", "Private Nonprofit", "Private For-Profit")

  print(summary_by_type)

  # Visualization
  plot_matrix <- t(as.matrix(
    data.frame(
      Completion = summary_by_type$C150_4_POOLED,
      Earnings_scaled = summary_by_type$MD_EARN_WNE_P10 / 100000
    )
  ))
  barplot(
    plot_matrix,
    beside = TRUE,
    names.arg = summary_by_type$CONTROL_NAME,
    legend.text = c("Completion Rate", "Earnings (/100k)"),
    args.legend = list(x = "topleft", bty = "n"),
    col = c("skyblue", "orange"),
    main = "Avg Completion and Earnings by Institution Type",
    ylab = "Value (Rate or $/100k)"
  )
  grid(nx = NA, ny = NULL)
}
```

```
##   CONTROL C150_4_POOLED MD_EARN_WNE_P10       CONTROL_NAME
## 1       1     0.4679681        51568.05             Public
## 2       2     0.5682953        54686.42  Private Nonprofit
## 3       3     0.4578916        46070.35 Private For-Profit
```



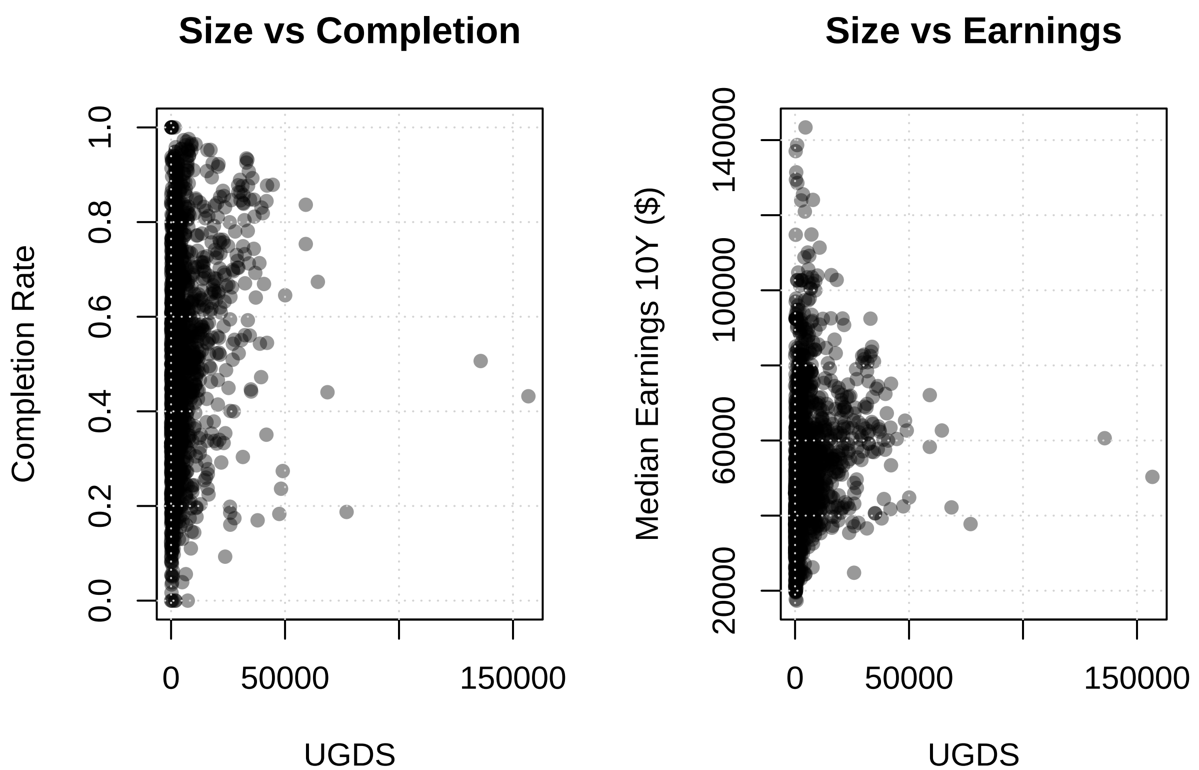
---

20) Size vs Performance

```r
need2 <- c("UGDS","C150_4_POOLED","MD_EARN_WNE_P10")
if (all(need2 %in% names(df))) {
  size_df <- data.frame(
    UGDS=suppressWarnings(as.numeric(df$UGDS)),
    C150_4_POOLED=suppressWarnings(as.numeric(df$C150_4_POOLED)),
    MD_EARN_WNE_P10=suppressWarnings(as.numeric(df$MD_EARN_WNE_P10))
  )
  size_df <- size_df[ complete.cases(size_df), ]
  corr_size_completion <- cor(size_df$UGDS, size_df$C150_4_POOLED)
  corr_size_earnings <- cor(size_df$UGDS, size_df$MD_EARN_WNE_P10)
  cat(sprintf("Corr(UGDS, Completion): %.3f\n", corr_size_completion))
  cat(sprintf("Corr(UGDS, Earnings): %.3f\n", corr_size_earnings))
}
```

```
## Corr(UGDS, Completion): 0.151
## Corr(UGDS, Earnings): 0.169
```

```r
  par(mfrow=c(1,2))
  plot(size_df$UGDS, size_df$C150_4_POOLED, pch=16, col=rgb(0,0,0,0.4),
       xlab="UGDS", ylab="Completion Rate", main="Size vs Completion"); grid()
  plot(size_df$UGDS, size_df$MD_EARN_WNE_P10, pch=16, col=rgb(0,0,0,0.4),
       xlab="UGDS", ylab="Median Earnings 10Y ($)", main="Size vs Earnings"); grid()
```



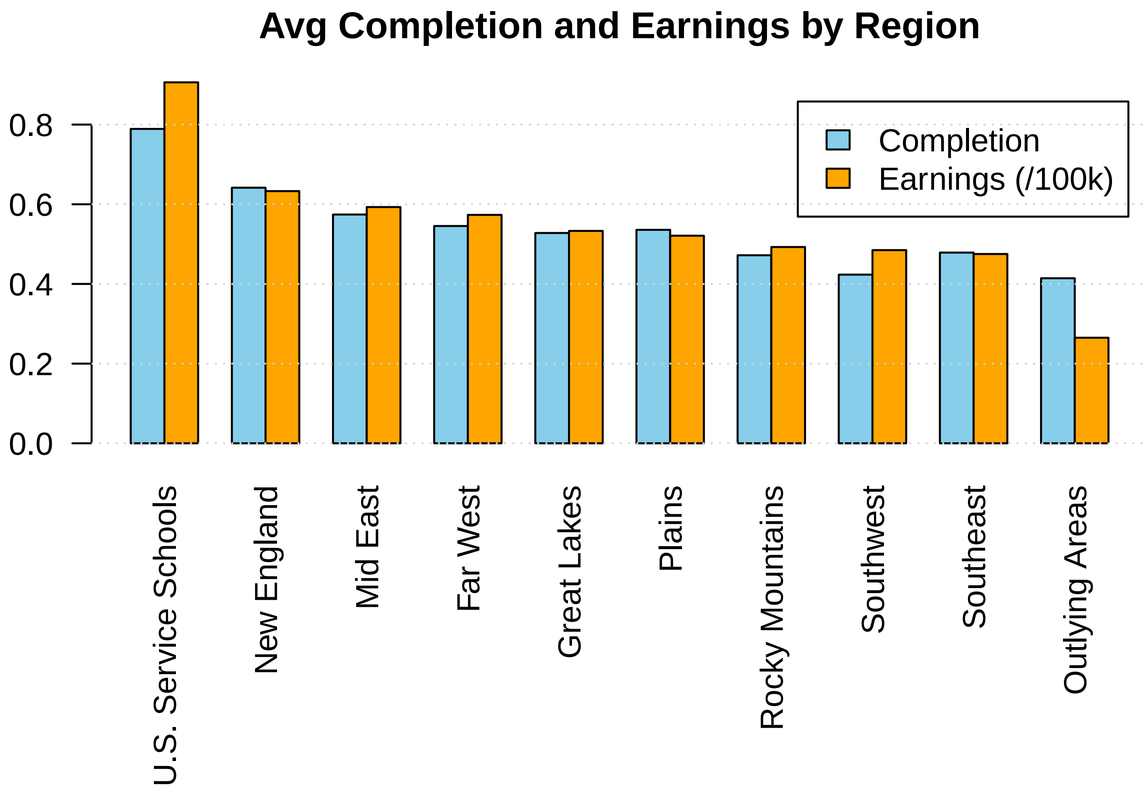

```
par(mfrow=c(1,1))
```

---

21) Region Summary (Bars) & Success Index

```
if (all(c("REGION","C150_4_POOLED","MD_EARN_WNE_P10") %in% names(df))) {
  region_df <- data.frame(
    REGION = suppressWarnings(as.numeric(df$REGION)),
    C150_4_POOLED = suppressWarnings(as.numeric(df$C150_4_POOLED)),
    MD_EARN_WNE_P10 = suppressWarnings(as.numeric(df$MD_EARN_WNE_P10))
  )
  region_df <- region_df[ complete.cases(region_df), ]
  region_df$REGION_NAME <- region_map[ as.character(region_df$REGION) ]

  summary_by_region <- aggregate(cbind(C150_4_POOLED, MD_EARN_WNE_P10) ~ REGION_NAME, region_df, mean)
  summary_by_region <- summary_by_region[ order(-summary_by_region$MD_EARN_WNE_P10), ]
  summary_by_region

  # Bars (earnings scaled)
  par(mar=c(10,4,3,1))
  mat <- t(as.matrix(cbind(summary_by_region$C150_4_POOLED, summary_by_region$MD_EARN_WNE_P10/100000)))
  barplot(mat, beside=TRUE, names.arg=summary_by_region$REGION_NAME, las=2,
          col=c("skyblue","orange"), legend.text=c("Completion","Earnings (/100k)"),
          main="Avg Completion and Earnings by Region")
  grid(nx=NA, ny=NULL)

  # Success Index (normalize columns 0-1 and average)
  rng_norm <- function(v) (v - min(v, na.rm=TRUE)) / (max(v, na.rm=TRUE) - min(v, na.rm=TRUE))
```

```
  summary_by_region$completion_norm <- rng_norm(summary_by_region$C150_4_POOLED)
  summary_by_region$earnings_norm   <- rng_norm(summary_by_region$MD_EARN_WNE_P10)
  summary_by_region$success_index   <- (summary_by_region$completion_norm + summary_by_region$earnings_
  summary_by_region <- summary_by_region[ order(-summary_by_region$success_index), ]
}
```



---

16) Field-of-Study (CIP) Area Summaries

```
degree_column_name <- "CREDLEV_NAME"

# Replace privacy strings with NA
repl <- function(x) {
  x[x %in% c("PS", "PrivacySuppressed", "NULL", "None", "")] <- NA
  x
}
for (nm in names(df_field)) {
  df_field[[nm]] <- repl(df_field[[nm]])
}

# convert key columns to numeric
num_cols <- intersect(c(
  "EARN_MDN_1YR",
  "EARN_MDN_4YR",
  "EARN_GT_THRESHOLD_1YR",
  "EARN_GT_THRESHOLD_5YR",
  "DEBT_MDN_SUPP",
```

```r
  "NET_PRICE"
), names(df_field))
for (cname in num_cols) {
  df_field[[cname]] <- suppressWarnings(as.numeric(df_field[[cname]]))
}

# map detailed cip codes to broader subject areas
if ("CIPCODE" %in% names(df_field)) {
  df_field$CIP2 <- suppressWarnings(as.integer(as.numeric(df_field$CIPCODE) %/% 100))
  cip2_map <- c(
    "1" = "Agriculture & Related Sciences",
    "3" = "Natural Resources & Conservation",
    "4" = "Architecture & Related Services",
    "9" = "Communications & Journalism",
    "10" = "Communications Technologies",
    "11" = "Computer & Information Sciences",
    "13" = "Education",
    "14" = "Engineering",
    "15" = "Engineering Technologies",
    "19" = "Family & Consumer Sciences",
    "22" = "Legal Studies",
    "24" = "Liberal Arts & Humanities",
    "27" = "Mathematics & Statistics",
    "30" = "Multi/Interdisciplinary Studies",
    "38" = "Philosophy & Religious Studies",
    "40" = "Physical Sciences",
    "42" = "Psychology",
    "44" = "Public Administration & Social Services",
    "45" = "Social Sciences",
    "50" = "Visual & Performing Arts",
    "51" = "Health Professions & Related Programs",
    "52" = "Business, Management, & Marketing",
    "12" = "Personal & Culinary Services",
    "23" = "English Language & Literature",
    "31" = "Parks/Rec/Leisure & Kinesiology",
    "39" = "Theology & Religious Vocations",
    "46" = "Construction Trades",
    "48" = "Precision Production",
    "54" = "History"
  )
  df_field$CIPAREA <- cip2_map[as.character(df_field$CIP2)]
}

# map numeric credential levels to text names
credlev_map <- c(
  "2" = "Associate's",
  "3" = "Bachelor's",
  "5" = "Master's",
  "6" = "Doctoral",
  "7" = "First Professional",
  "1" = "Certificate"
)
df_field$CREDLEV_NAME <- credlev_map[as.character(df_field$CREDLEV)]
```

```r
# filter for only the assc., bach, mast. degree levels
desired_degrees <- c("Associate's", "Bachelor's", "Master's")
df_filtered <- df_field[df_field[[degree_column_name]] %in% desired_degrees, ]

# calculate median earnings by area and degree
summary_cip_degree <- aggregate(df_filtered["EARN_MDN_1YR"],
  by = list(
    CIPAREA = df_filtered$CIPAREA,
    Degree = df_filtered[[degree_column_name]]
  ),
  FUN = median,
  na.rm = TRUE
)

# reshape data from long to wide format
earnings_by_degree <- reshape(summary_cip_degree,
  idvar = "CIPAREA",
  timevar = "Degree",
  direction = "wide"
)

# sort by bachelor's degree earnings and display
earnings_by_degree_sorted <- earnings_by_degree[order(earnings_by_degree$`EARN_MDN_1YR.Bachelor's`, dec
earnings_by_degree_sorted
```

```
##                                    CIPAREA EARN_MDN_1YR.Associate's
## 7                      Construction Trades                  43251.5
## 9                              Engineering                  48263.0
## 10                Engineering Technologies                  52108.0
## 13   Health Professions & Related Programs                  53558.0
## 6          Computer & Information Sciences                  39348.0
## 17                Mathematics & Statistics                  16986.0
## 3        Business, Management, & Marketing                  34644.0
## 2          Architecture & Related Services                  42331.0
## 8                                Education                  24414.0
## 23                       Physical Sciences                  26948.0
## 1           Agriculture & Related Sciences                  36090.0
## 15                          Legal Studies                   34421.0
## 26 Public Administration & Social Services                  31677.0
## 27                         Social Sciences                  27041.0
## 16               Liberal Arts & Humanities                  27221.5
## 18          Multi/Interdisciplinary Studies                 29638.0
## 4             Communications & Journalism                   26437.0
## 19         Natural Resources & Conservation                 32567.0
## 12              Family & Consumer Sciences                  26593.0
## 21             Personal & Culinary Services                 26692.0
## 28           Theology & Religious Vocations                 29571.0
## 25                              Psychology                  26232.5
## 14                                 History                       NA
## 20          Parks/Rec/Leisure & Kinesiology                 27493.5
## 22           Philosophy & Religious Studies                 34279.5
## 11            English Language & Literature                 26148.5
## 29               Visual & Performing Arts                   23694.0
```

```
## 5             Communications Technologies                 23625.5
## 24                      Precision Production                 41017.0
##    EARN_MDN_1YR.Bachelor's EARN_MDN_1YR.Master's
## 7                  72864.5                    NA
## 9                  72087.0               92940.0
## 10                 64339.0               95527.0
## 13                 61265.0               69106.5
## 6                  61234.5               87667.0
## 17                 51034.5               80690.0
## 3                  49163.0               71067.0
## 2                  46449.0               60017.0
## 8                  42150.0               55793.5
## 23                 41810.5               65799.0
## 1                  41077.5               57773.0
## 15                 38432.0               73893.0
## 26                 37990.0               53515.0
## 27                 36504.0               61019.0
## 16                 36340.0               51346.0
## 18                 36206.0               57257.0
## 4                  35160.5               54196.0
## 19                 34321.0               54909.0
## 12                 33892.0               49604.0
## 21                 32775.0               21370.0
## 28                 32202.0               47762.0
## 25                 31870.5               51090.0
## 14                 31217.0               44930.0
## 20                 31013.0               44749.0
## 22                 29861.0               53160.0
## 11                 29605.0               43520.0
## 29                 25103.5               32650.0
## 5                  24892.0               41411.0
## 24                 19151.0                    NA
```

```
# Master's earnings multiplier
earnings_by_degree_sorted$Masters_Multiplier <- earnings_by_degree_sorted$`EARN_MDN_1YR.Master's` / earn

# raw earnings gain
earnings_by_degree_sorted$Masters_Raw_Gain <- earnings_by_degree_sorted$`EARN_MDN_1YR.Master's` - earnin

# Filter where multiplier or raw gain not calcualted
combined_data <- earnings_by_degree_sorted[
  !is.na(earnings_by_degree_sorted$Masters_Multiplier) &
  is.finite(earnings_by_degree_sorted$Masters_Multiplier) &
  !is.na(earnings_by_degree_sorted$Masters_Raw_Gain),
]

# descending order
combined_sorted <- combined_data[order(combined_data$Masters_Multiplier, decreasing = TRUE), ]


final_table <- data.frame(
  Discipline = combined_sorted$CIPAREA,
  Masters_Earnings_Multiplier = combined_sorted$Masters_Multiplier,
  Masters_Earnings_Gain = combined_sorted$Masters_Raw_Gain, # corrected column name
```

```r
  Bachelors_Median_Earnings = combined_sorted$`EARN_MDN_1YR.Bachelor's`,
  Masters_Median_Earnings = combined_sorted$`EARN_MDN_1YR.Master's`
)

print(final_table)
```

```
##                                 Discipline Masters_Earnings_Multiplier
## 1                           Legal Studies                   1.9226946
## 2            Philosophy & Religious Studies                   1.7802485
## 3                          Social Sciences                   1.6715702
## 4              Communications Technologies                   1.6636269
## 5                               Psychology                   1.6030498
## 6          Natural Resources & Conservation                   1.5998660
## 7            Multi/Interdisciplinary Studies                   1.5814230
## 8                   Mathematics & Statistics                   1.5810873
## 9                        Physical Sciences                   1.5737434
## 10             Communications & Journalism                   1.5413888
## 11                 Engineering Technologies                   1.4847449
## 12           Theology & Religious Vocations                   1.4831998
## 13             English Language & Literature                   1.4700220
## 14                Family & Consumer Sciences                   1.4635902
## 15        Business, Management, & Marketing                   1.4455383
## 16           Parks/Rec/Leisure & Kinesiology                   1.4429110
## 17                                  History                   1.4392799
## 18           Computer & Information Sciences                   1.4316603
## 19                 Liberal Arts & Humanities                   1.4129334
## 20 Public Administration & Social Services                   1.4086602
## 21              Agriculture & Related Sciences                   1.4064390
## 22                                Education                   1.3236892
## 23                  Visual & Performing Arts                   1.3006155
## 24           Architecture & Related Services                   1.2921053
## 25                              Engineering                   1.2892755
## 26    Health Professions & Related Programs                   1.1279931
## 27              Personal & Culinary Services                   0.6520214
##    Masters_Earnings_Gain Bachelors_Median_Earnings Masters_Median_Earnings
## 1                35461.0                   38432.0                 73893.0
## 2                23299.0                   29861.0                 53160.0
## 3                24515.0                   36504.0                 61019.0
## 4                16519.0                   24892.0                 41411.0
## 5                19219.5                   31870.5                 51090.0
## 6                20588.0                   34321.0                 54909.0
## 7                21051.0                   36206.0                 57257.0
## 8                29655.5                   51034.5                 80690.0
## 9                23988.5                   41810.5                 65799.0
## 10               19035.5                   35160.5                 54196.0
## 11               31188.0                   64339.0                 95527.0
## 12               15560.0                   32202.0                 47762.0
## 13               13915.0                   29605.0                 43520.0
## 14               15712.0                   33892.0                 49604.0
## 15               21904.0                   49163.0                 71067.0
## 16               13736.0                   31013.0                 44749.0
## 17               13713.0                   31217.0                 44930.0
## 18               26432.5                   61234.5                 87667.0
## 19               15006.0                   36340.0                 51346.0
```

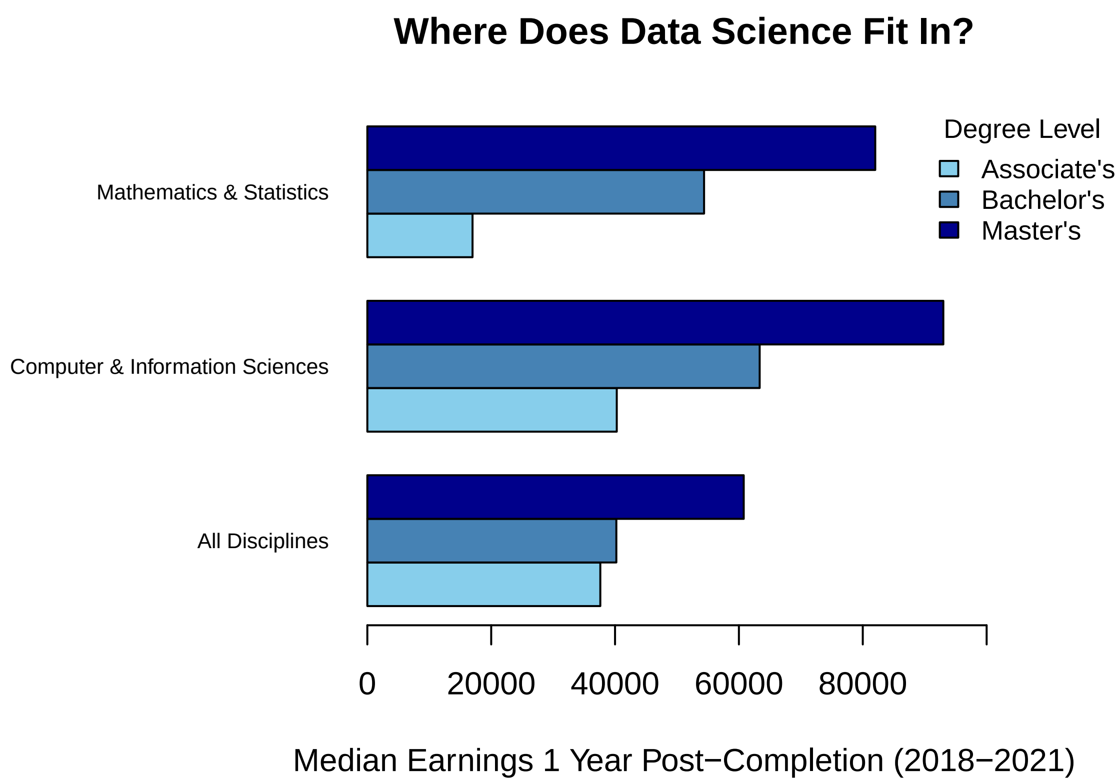
```
## 20           15525.0              37990.0             53515.0
## 21           16695.5              41077.5             57773.0
## 22           13643.5              42150.0             55793.5
## 23            7546.5              25103.5             32650.0
## 24           13568.0              46449.0             60017.0
## 25           20853.0              72087.0             92940.0
## 26            7841.5              61265.0             69106.5
## 27          -11405.0              32775.0             21370.0
```

---

22) Do Higher Degrees Lead to Higher Pay? —

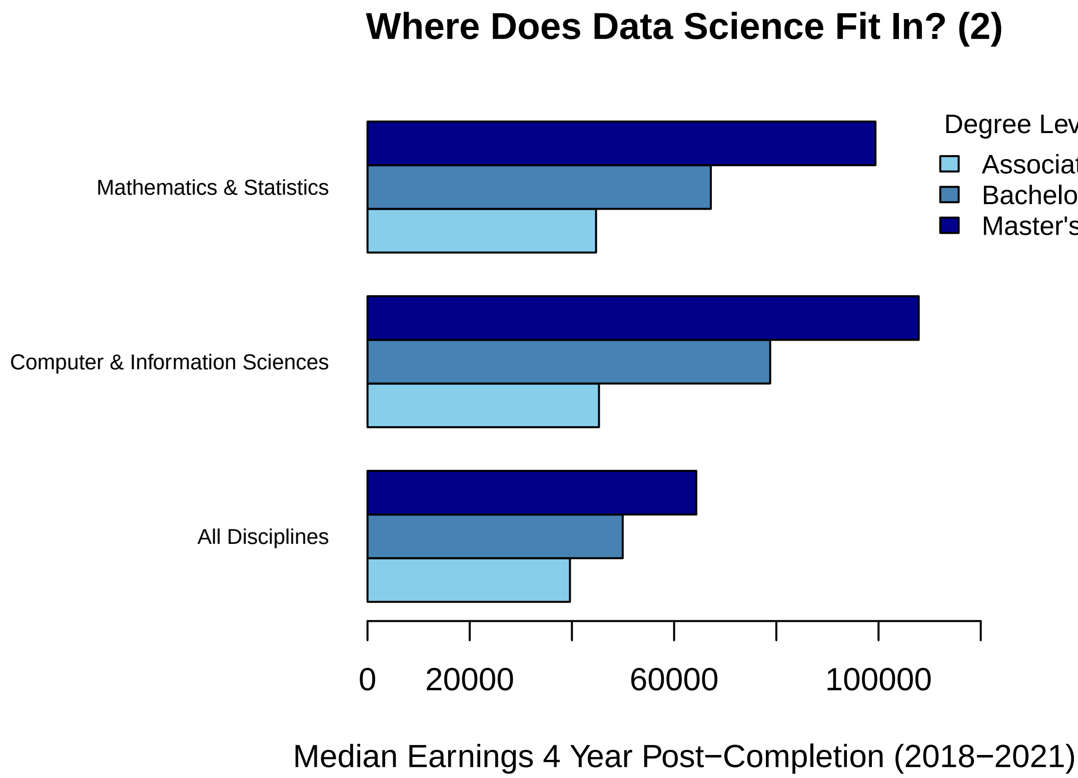
```
df_field$CREDLEV_NAME <- credlev_map[as.character(df_field$CREDLEV)]

# filter for only the degrees we want to compare
df_cred_compare <- df_field[df_field$CREDLEV %in% c(2, 3, 5), ]

# calculate mean earnings by discipline and overall median earnings
earnings_by_cred <- aggregate(EARN_MDN_1YR ~ CIPAREA + CREDLEV_NAME, data = df_cred_compare, FUN = mean

median_earnings_all <- aggregate(EARN_MDN_1YR ~ CREDLEV_NAME, data = df_cred_compare, FUN = median, na.

# create a baseline 'all disciplines' vector for comparison
control_data <- c(
  "Associate's" = median_earnings_all$EARN_MDN_1YR[median_earnings_all$CREDLEV_NAME == "Associate's"],
  "Bachelor's"  = median_earnings_all$EARN_MDN_1YR[median_earnings_all$CREDLEV_NAME == "Bachelor's"],
  "Master's"    = median_earnings_all$EARN_MDN_1YR[median_earnings_all$CREDLEV_NAME == "Master's"]
)

# data sci as comp + stats
datasci_cip <- c("Computer & Information Sciences", "Mathematics & Statistics")
plot_data_long <- earnings_by_cred[earnings_by_cred$CIPAREA %in% datasci_cip, ]

# manually separate, sort, and reshape data into a matrix for plotting
assoc_data <- plot_data_long[plot_data_long$CREDLEV_NAME == "Associate's", ]
bach_data  <- plot_data_long[plot_data_long$CREDLEV_NAME == "Bachelor's", ]
mast_data  <- plot_data_long[plot_data_long$CREDLEV_NAME == "Master's", ]

assoc_data <- assoc_data[order(assoc_data$CIPAREA), ]
bach_data  <- bach_data[order(bach_data$CIPAREA), ]
mast_data  <- mast_data[order(mast_data$CIPAREA), ]

plot_matrix <- cbind(
  "Associate's" = assoc_data$EARN_MDN_1YR,
  "Bachelor's"  = bach_data$EARN_MDN_1YR,
  "Master's"    = mast_data$EARN_MDN_1YR
)
rownames(plot_matrix) <- assoc_data$CIPAREA

# add the 'all disciplines' baseline row to the plot matrix
plot_matrix_with_control <- rbind("All Disciplines" = control_data, plot_matrix)
```

```r
# disable scientific notation
options(scipen = 999)

par(mar = c(5, 10, 4, 6))
max_val <- max(plot_matrix_with_control, na.rm = TRUE) * 1.1

# grouped horizontal barplot
barplot(
  height = t(plot_matrix_with_control),
  beside = TRUE,
  horiz = TRUE,
  main = "Where Does Data Science Fit In?",
  xlab = "Median Earnings 1 Year Post-Completion (2018-2021)",
  las = 1,
  cex.names = 0.7,
  col = c("skyblue", "steelblue", "darkblue"),
  xlim = c(0, max_val)
)

# uncrop the plot
par(xpd=TRUE)


legend(
  "topright",
  inset = c(-0.2, 0),
  legend = c("Associate's", "Bachelor's", "Master's"),
  fill = c("skyblue", "steelblue", "darkblue"),
  title = "Degree Level",
  bty = "n",
  cex = 0.8
)
```




```r
# same as prev code change EARN_MDN_4YR to EARN_MDN_1YR
earnings_by_cred <- aggregate(EARN_MDN_4YR ~ CIPAREA + CREDLEV_NAME, data = df_cred_compare, FUN = mean
median_earnings_all <- aggregate(EARN_MDN_4YR ~ CREDLEV_NAME, data = df_cred_compare, FUN = median, na.
control_data <- c(
  "Associate's" = median_earnings_all$EARN_MDN_4YR[median_earnings_all$CREDLEV_NAME == "Associate's"],
  "Bachelor's"  = median_earnings_all$EARN_MDN_4YR[median_earnings_all$CREDLEV_NAME == "Bachelor's"],
  "Master's"    = median_earnings_all$EARN_MDN_4YR[median_earnings_all$CREDLEV_NAME == "Master's"]
)

# select data science-related disciplines to plot
datasci_cip <- c("Computer & Information Sciences", "Mathematics & Statistics")
plot_data_long <- earnings_by_cred[earnings_by_cred$CIPAREA %in% datasci_cip, ]

# manually separate, sort, and reshape data into a matrix for plotting
assoc_data <- plot_data_long[plot_data_long$CREDLEV_NAME == "Associate's", ]
bach_data  <- plot_data_long[plot_data_long$CREDLEV_NAME == "Bachelor's", ]
mast_data  <- plot_data_long[plot_data_long$CREDLEV_NAME == "Master's", ]

assoc_data <- assoc_data[order(assoc_data$CIPAREA), ]
bach_data  <- bach_data[order(bach_data$CIPAREA), ]
mast_data  <- mast_data[order(mast_data$CIPAREA), ]

plot_matrix <- cbind(
  "Associate's" = assoc_data$EARN_MDN_4YR,
  "Bachelor's"  = bach_data$EARN_MDN_4YR,
  "Master's"    = mast_data$EARN_MDN_4YR
)
rownames(plot_matrix) <- assoc_data$CIPAREA
```

```r
plot_matrix_with_control <- rbind("All Disciplines" = control_data, plot_matrix)


# sisable scientific notation
options(scipen = 999)

par(mar = c(5, 10, 4, 6))

max_val <- max(plot_matrix_with_control, na.rm = TRUE) * 1.15

barplot(
  height = t(plot_matrix_with_control),
  beside = TRUE,
  horiz = TRUE,
  main = "Where Does Data Science Fit In? (2)",
  xlab = "Median Earnings 4 Year Post-Completion (2018-2021)",
  las = 1,
  cex.names = 0.7,
  col = c("skyblue", "steelblue", "darkblue"),
  xlim = c(0, max_val)
)

# uncrop the plot
par(xpd=TRUE)

legend(
  "topright",
  inset = c(-0.2, 0),
  legend = c("Associate's", "Bachelor's", "Master's"),
  fill = c("skyblue", "steelblue", "darkblue"),
  title = "Degree Level",
  bty = "n",
  cex = 0.8
)
```


Where Does Data Science Fit In? (2)
Degree Level
Associate's
Bachelor's
Master's
Mathematics & Statistics
Computer & Information Sciences
All Disciplines
0
20000
60000
100000
Median Earnings 4 Year Post−Completion (2018−2021)