Three Super Refined Graphs

Ben Goldstone

9/29/2023

College Distance Data Import

College Distance Dataset

```
collegeDistance <- read_csv("~/CSVs/CollegeDistance.csv")
```

```
## New names:
## * `` -> ...1
```

```
## Rows: 4739 Columns: 15
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): gender, ethnicity, fcollege, mcollege, home, urban, income, region
## dbl (7): ...1, score, unemp, wage, distance, tuition, education
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(collegeDistance)
```

```
## # A tibble: 6 x 15
##    ...1 gender ethnicity score fcollege mcollege home  urban unemp  wage
##   <dbl> <chr>  <chr>     <dbl> <chr>    <chr>    <chr> <chr> <dbl> <dbl>
## 1     1 male   other      39.2 yes      no       yes   yes    6.20  8.09
## 2     2 female other      48.9 no       no       yes   yes    6.20  8.09
## 3     3 male   other      48.7 no       no       yes   yes    6.20  8.09
## 4     4 male   afam       40.4 no       no       yes   yes    6.20  8.09
## 5     5 female other      40.5 no       no       no    yes    5.60  8.09
## 6     6 male   other      54.7 no       no       yes   yes    5.60  8.09
## # ... with 5 more variables: distance <dbl>, tuition <dbl>, education <dbl>,
## #   income <chr>, region <chr>
```

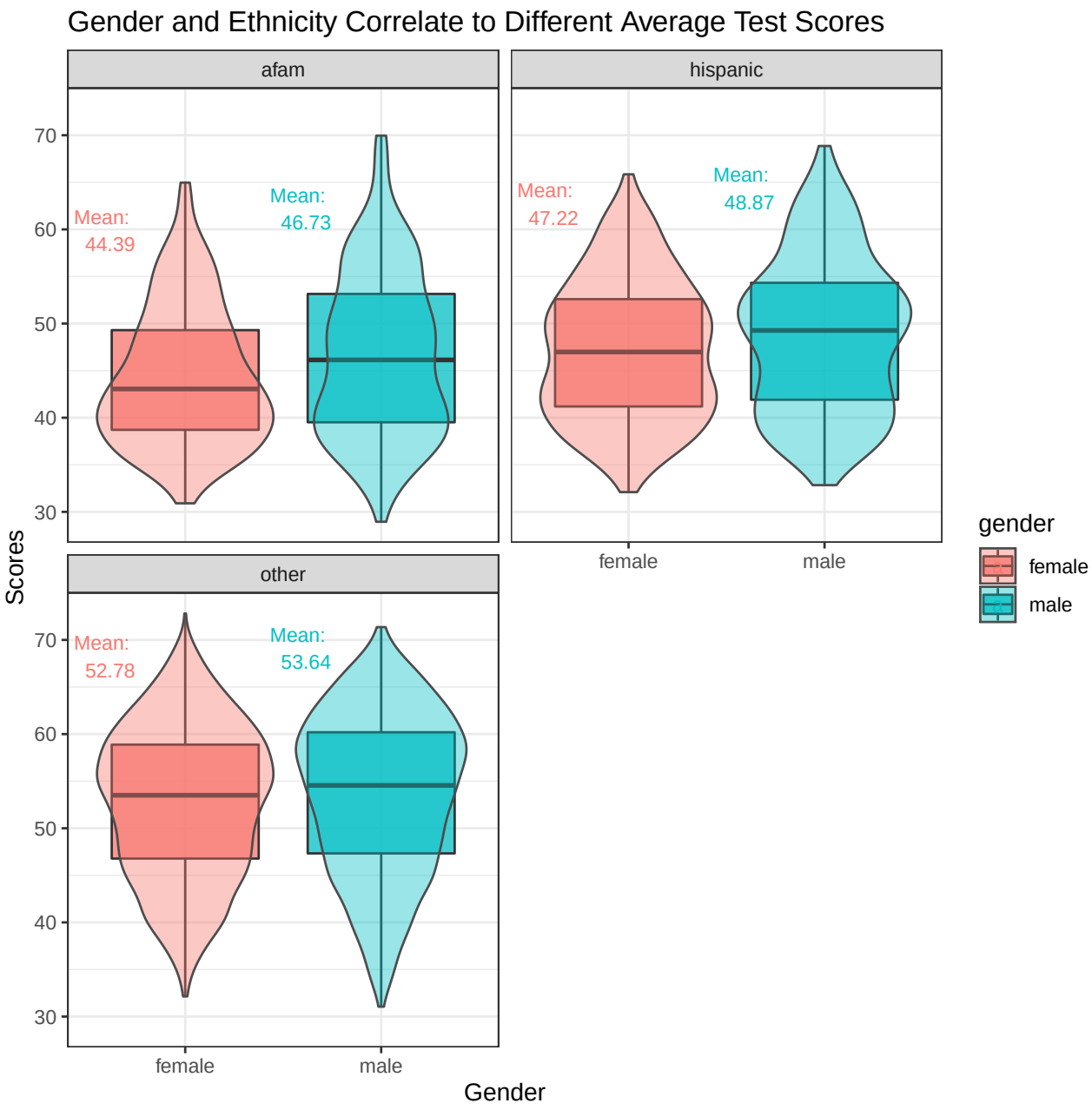
```
library(dplyr)
means = data.frame(collegeDistance %>%
  group_by(gender, ethnicity) %>%
  summarise(mean_score = mean(score)))
```

```
## `summarise()` has grouped output by 'gender'. You can override using the `.groups` argument.
```

Composite test scores and gender

```
gf_boxplot(score ~ gender,data = collegeDistance, fill =  ~ gender,alpha = 0.75) %>%
  gf_facet_wrap( ~ ethnicity, nrow = 3, ncol = 2) %>%
  gf_labs(title="Gender and Ethnicity Correlate to Different Average Test Scores",
```

```
  x="Gender",y="Scores") +
geom_violin(alpha = 0.4, color = "grey30") +
geom_text(data = means, aes(x = gender, y = mean_score, label = sprintf("Mean:\n%.2f",
  mean_score), color = gender),
  position = position_dodge(width = 0.8), vjust=-3, hjust=2, size=3)
```



China Income Data Import

China Income Dataset

```
ChinaIncome <- read_csv("~/CSVs/ChinaIncome.csv")
```

```
## Rows: 37 Columns: 6
```

```
## -- Column specification --------------------------------------------------------
```

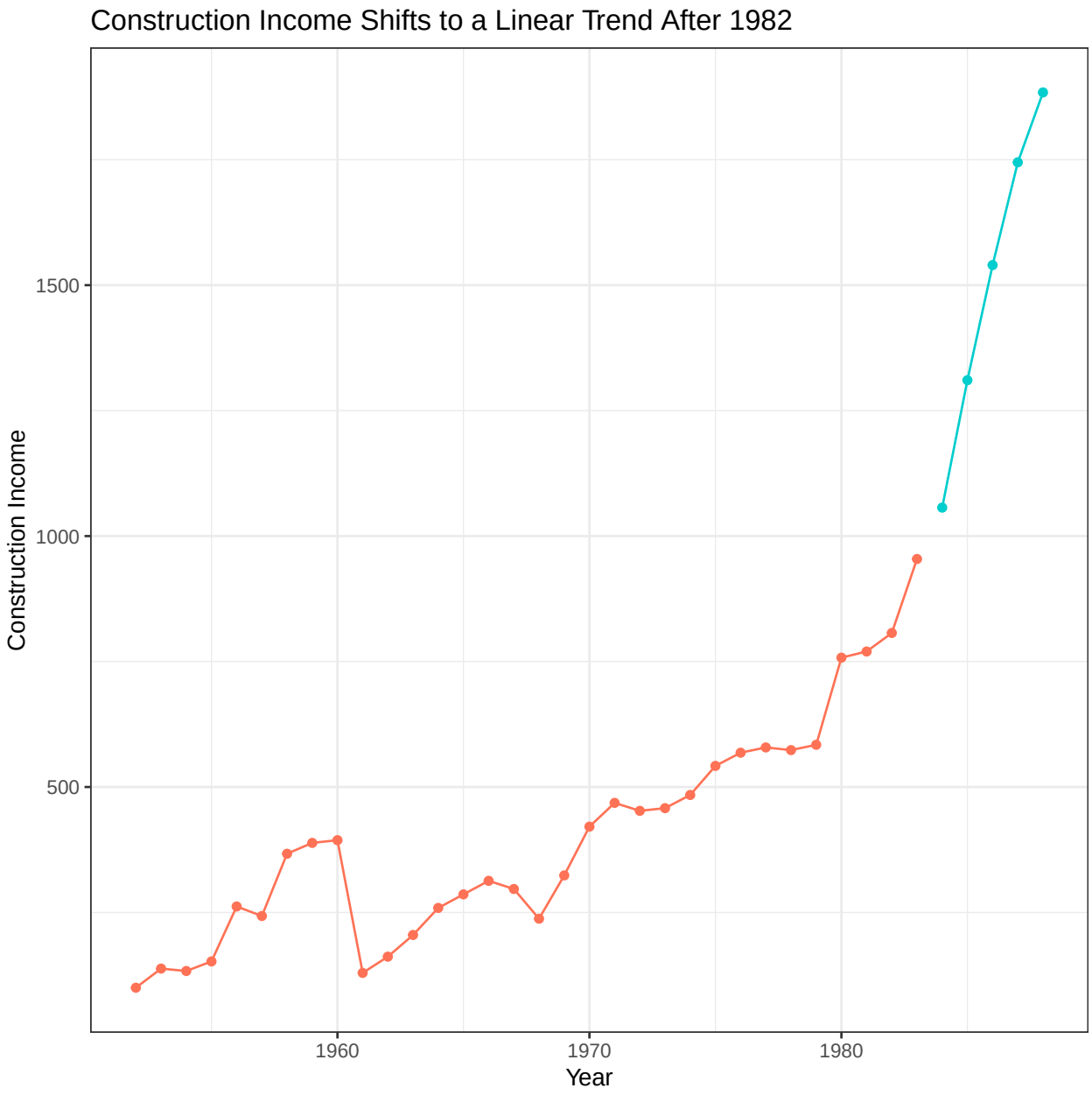
```
## Delimiter: ","
## dbl (6): year, agricultureIncome, commerceIncome, constructionIncome, indust...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(ChinaIncome)
```

```
## # A tibble: 6 x 6
##    year agricultureIncome commerceIncome constructionIncome industryIncome
##   <dbl>             <dbl>          <dbl>              <dbl>          <dbl>
## 1  1952               100            100                100            100
## 2  1953               102.           133                138.           134.
## 3  1954               103.           136.               133.           159.
## 4  1955               112.           138.               152.           169.
## 5  1956               116.           147.               262.           219.
## 6  1957               120.           147.               243.           244.
## # ... with 1 more variable: transportIncome <dbl>
```

Construction Income over Time

```
const_income_color_sceme = c("TRUE" = "cyan3", "FALSE" = "coral1")
const_income_above_1000 = subset(ChinaIncome,constructionIncome > 1000)
const_income_below_1000 = subset(ChinaIncome,constructionIncome <= 1000)
gf_point(constructionIncome~year,data=ChinaIncome,color = ~ (constructionIncome > 1000)) %>%
  gf_line(constructionIncome~year, data=const_income_above_1000,
          color=const_income_color_sceme["TRUE"]) %>%
  gf_line(constructionIncome~year, data=const_income_below_1000,
          color=const_income_color_sceme["FALSE"]) %>%
  gf_labs(title="Construction Income Shifts to a Linear Trend After 1982",
          y="Construction Income", x="Year") %>%
  gf_theme(legend.position="none") +
  # Sets colors based on True or False Condition
  scale_color_manual(
    values = const_income_color_sceme) +
  # Sets legend header
  labs(color="Construction Income")
```



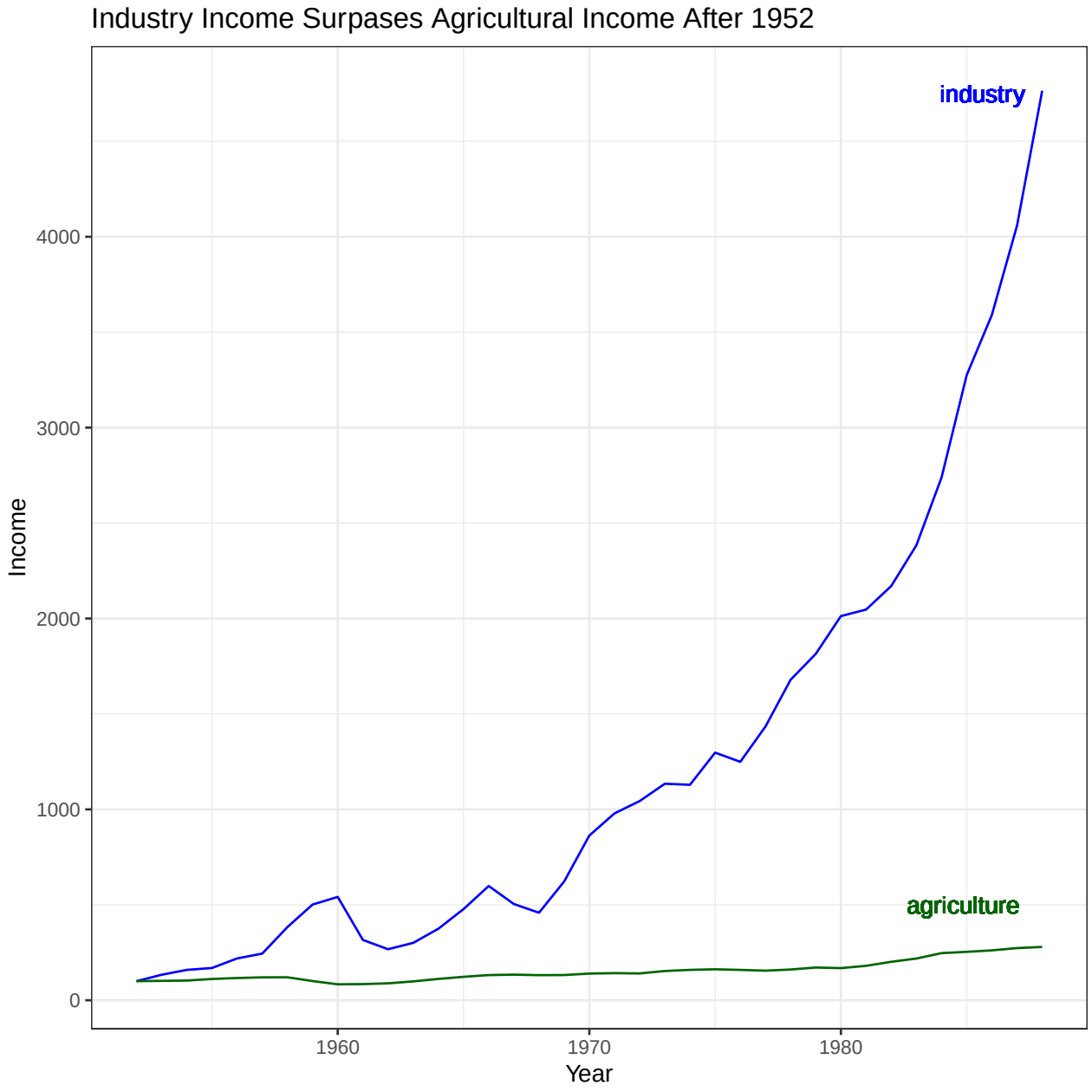
Industry Income and Agriculture Income

```
# Sets color for industries
industry = "blue"
agriculture="darkgreen"

# Plots lines for industry and agriculture layered
ind_agr_plot = gf_line(industryIncome~year,data=ChinaIncome,color=industry) %>%
  gf_line(agricultureIncome~year,data=ChinaIncome,color=agriculture) %>%

# Puts "Industry" and "Agriculture" text on plot with the corresponding color
# wish I could make the font thinner or more spread out...
gf_text(x=1988,y=4750,label="industry",color=industry,hjust = 1.2) %>%
  gf_text(x=1988,y=500,color=agriculture,label="agriculture", hjust=1.2) %>%
```

```
  gf_labs(title="Industry Income Surpases Agricultural Income After 1952",
          x="Year",y="Income")
ind_agr_plot
```




```
# ggplotly(ind_agr_plot)
```