AE 04: Pivoting Cornell Degrees

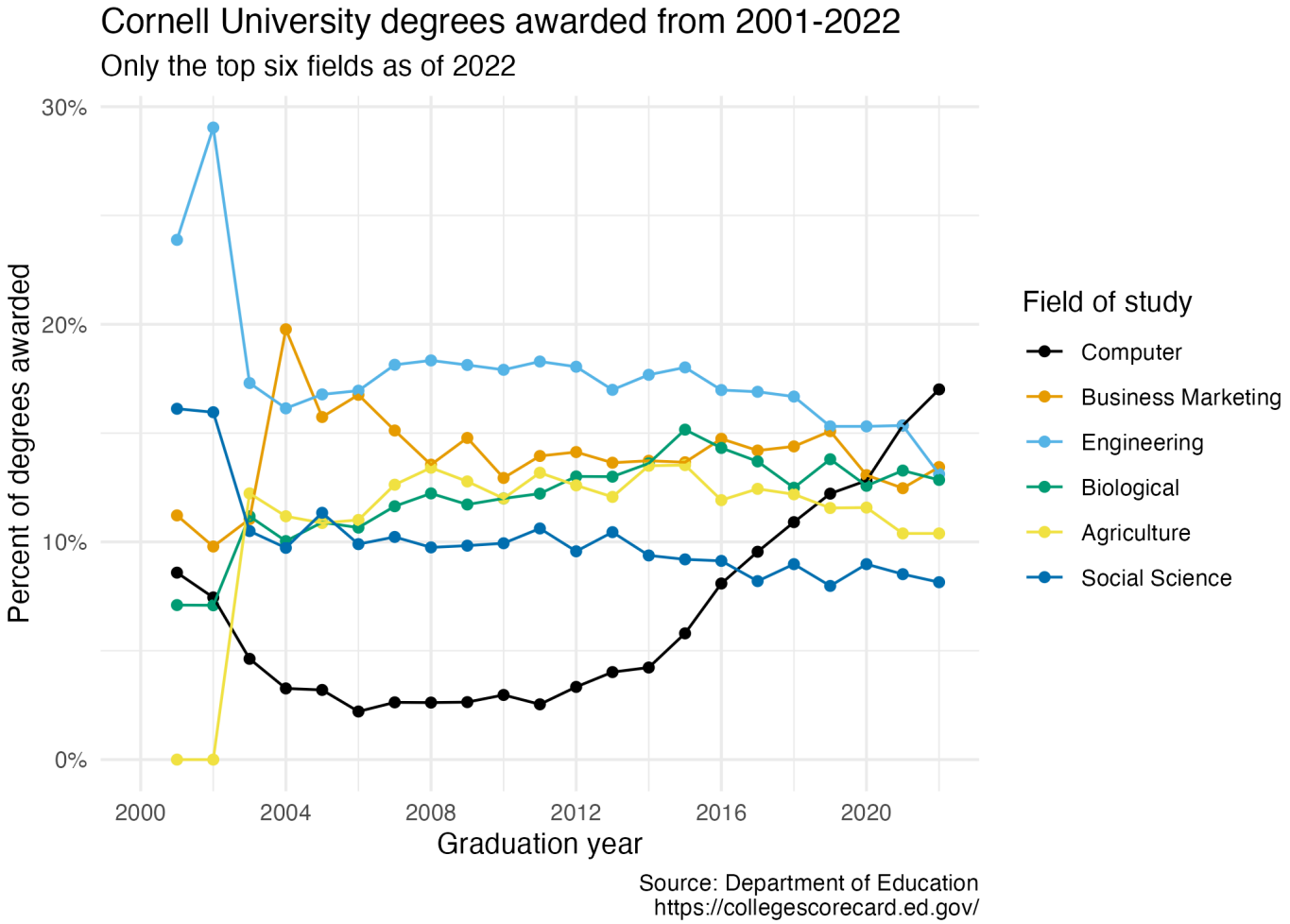
Suggested answers

APPLICATION EXERCISE ANSWERS

MODIFIED
September 17, 2024

Goal

Our ultimate goal in this application exercise is to make the following data visualization.



- **Your turn (3 minutes):** Take a close look at the plot and describe what it shows in 2-3 sentences.

Add your response here.

Data

The data come from the Department of Education's College Scorecard.

They make the data available through online dashboards and an API, but I've prepared the data for you in a CSV file. Let's load that in.

```
library(tidyverse)
library(scales)

cornell_deg <- read_csv("data/cornell-degrees.csv")
```

And let's take a look at the data.

```
cornell_deg
```

```
# A tibble: 6 × 23
  field_of_study  `2001` `2002` `2003` `2004` `2005` `2006` `2007` `2008` `2009`
  <chr>            <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
1 Computer        0.0859 0.0745 0.0463 0.0327  0.032 0.0221 0.0263 0.0262 0.0264
2 Business Marke… 0.112  0.0979 0.110  0.198   0.157 0.168  0.151  0.136  0.148
3 Engineering     0.239  0.290  0.173  0.161   0.168 0.170  0.181  0.183  0.181
4 Biological      0.071  0.0709 0.112  0.100   0.109 0.107  0.116  0.122  0.117
5 Agriculture     0      0      0.122  0.112   0.109 0.110  0.126  0.134  0.128
6 Social Science  0.161  0.160  0.105  0.0973  0.113 0.099  0.102  0.0975 0.0983
# ℹ 13 more variables: `2010` <dbl>, `2011` <dbl>, `2012` <dbl>, `2013` <dbl>,
#   `2014` <dbl>, `2015` <dbl>, `2016` <dbl>, `2017` <dbl>, `2018` <dbl>,
#   `2019` <dbl>, `2020` <dbl>, `2021` <dbl>, `2022` <dbl>
```

The dataset has 6 rows and 23 columns. The first column (variable) is the `field_of_study`, which are the 6 most frequent fields of study for students graduating in 2022.[1] The remaining columns show the proportion of degrees awarded in each year from 2001-2022.

[1] For the sake of application, I omitted the other 32 possible fields of study.

- **Your turn (4 minutes):** Take a look at the plot we aim to make and sketch the data frame we need to make the plot. Determine what each row and each column of the data frame should be. *Hint:* We need data to be in columns to map to `aes`thetic elements of the plot.
    - Columns: `year`, `pct` , `field_of_study`
    - Rows: Combination of year and field of study

> One row for each year and one column for each field of study

> Confused why we don't want one row for each year and one column for each field of study? See the appendix.

Pivoting

- **Demo:** Pivot the `cornell_deg` data frame *longer* such that each row represents a field of study / year combination and `year` and `pct` age of graduates for that year are columns in the data frame.

```
cornell_deg |>
  pivot_longer(
    cols = -field_of_study,
    names_to = "year",
    values_to = "pct"
  )
```

```
# A tibble: 132 × 3
   field_of_study year     pct
   <chr>          <chr>  <dbl>
 1 Computer       2001  0.0859
 2 Computer       2002  0.0745
 3 Computer       2003  0.0463
 4 Computer       2004  0.0327
 5 Computer       2005  0.032
 6 Computer       2006  0.0221
 7 Computer       2007  0.0263
 8 Computer       2008  0.0262
 9 Computer       2009  0.0264
10 Computer       2010  0.0297
# ℹ 122 more rows
```

- **Question:** What is the type of the `year` variable? Why? What should it be?

It's a character (`chr`) variable since the information came from the columns of the original data frame and R cannot know that these character strings represent years. The variable type should be numeric.

- **Demo:** Start over with pivoting, and this time also make sure `year` is a numerical variable in the resulting data frame.

```
cornell_deg |>
  pivot_longer(
    cols = -field_of_study,
    names_to = "year",
    names_transform = parse_number,
    values_to = "pct"
  )
```

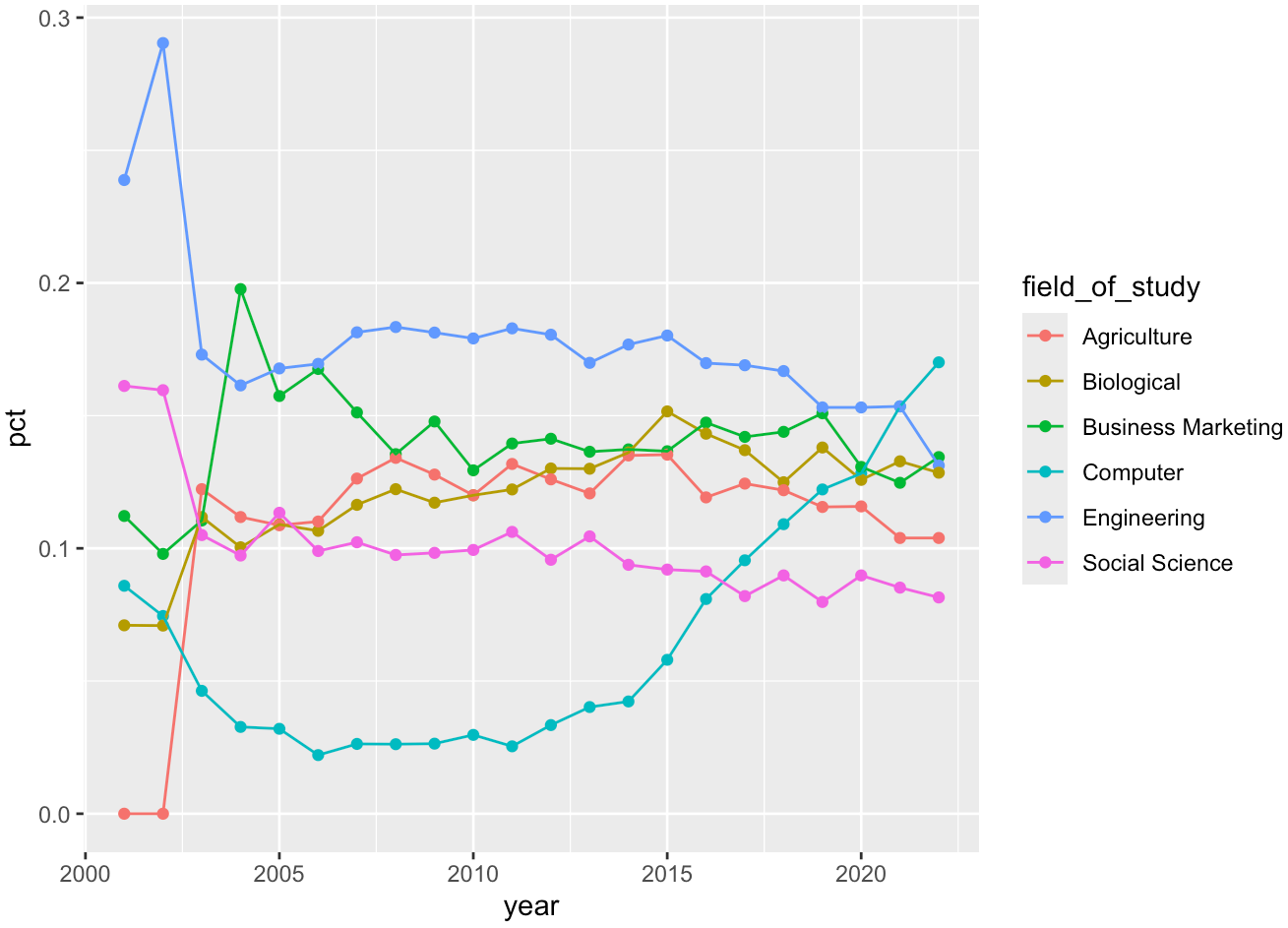
```
# A tibble: 132 × 3
   field_of_study  year    pct
```

```
   <chr>          <dbl>  <dbl>
 1 Computer        2001 0.0859
 2 Computer        2002 0.0745
 3 Computer        2003 0.0463
 4 Computer        2004 0.0327
 5 Computer        2005 0.032
 6 Computer        2006 0.0221
 7 Computer        2007 0.0263
 8 Computer        2008 0.0262
 9 Computer        2009 0.0264
10 Computer        2010 0.0297
# ℹ 122 more rows
```

Plotting

- **Your turn (5 minutes):** Now we start making our plot, but let's not get too fancy right away. Create the following plot, which will serve as the "first draft" on the way to our Goal. Do this by adding on to your pipeline from earlier.

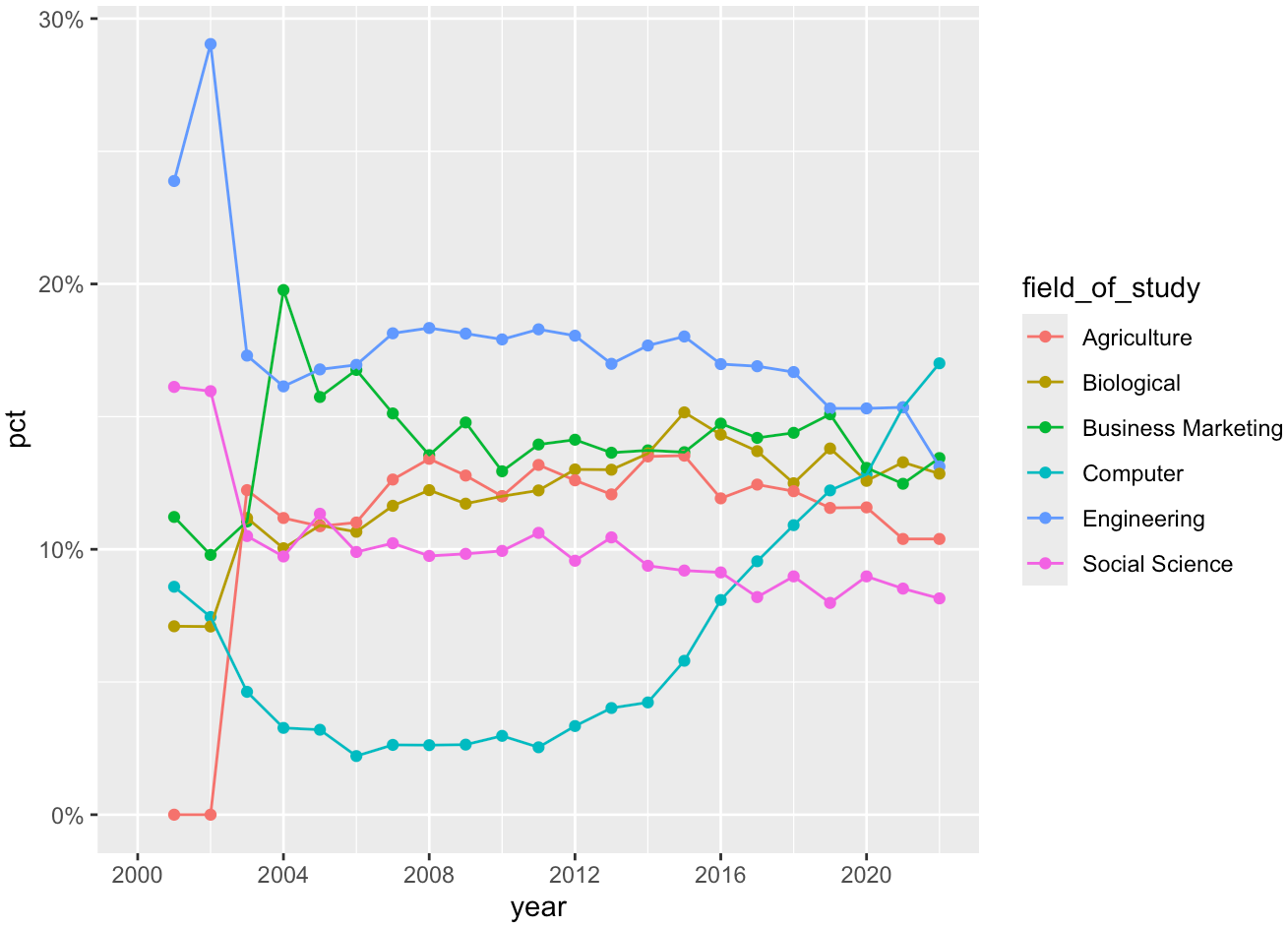
```
cornell_deg |>
  pivot_longer(
    cols = -field_of_study,
    names_to = "year",
    names_transform = parse_number,
    values_to = "pct"
  ) |>
  ggplot(aes(x = year, y = pct, color = field_of_study)) +
  geom_point() +
  geom_line()
```



- **Your turn (4 minutes):** What aspects of the plot need to be updated to go from the draft you created above to the Goal plot at the beginning of this application exercise.
  - x-axis scale: need to go from 2000 to 2022 in increments of 4 years
  - y-axis scale: percentage labeling
  - line colors
  - axis labels: title, subtitle, x, y, caption
  - theme
  - legend: position, order of values, and border
- **Demo:** Update x-axis scale such that the years displayed go from 2000 to 2022 in increments of 4 years. Update y-axis scale so it uses percentage formatting. Do this by adding on to your pipeline from earlier.

```
cornell_deg |>
  pivot_longer(
    cols = -field_of_study,
    names_to = "year",
    names_transform = parse_number,
```

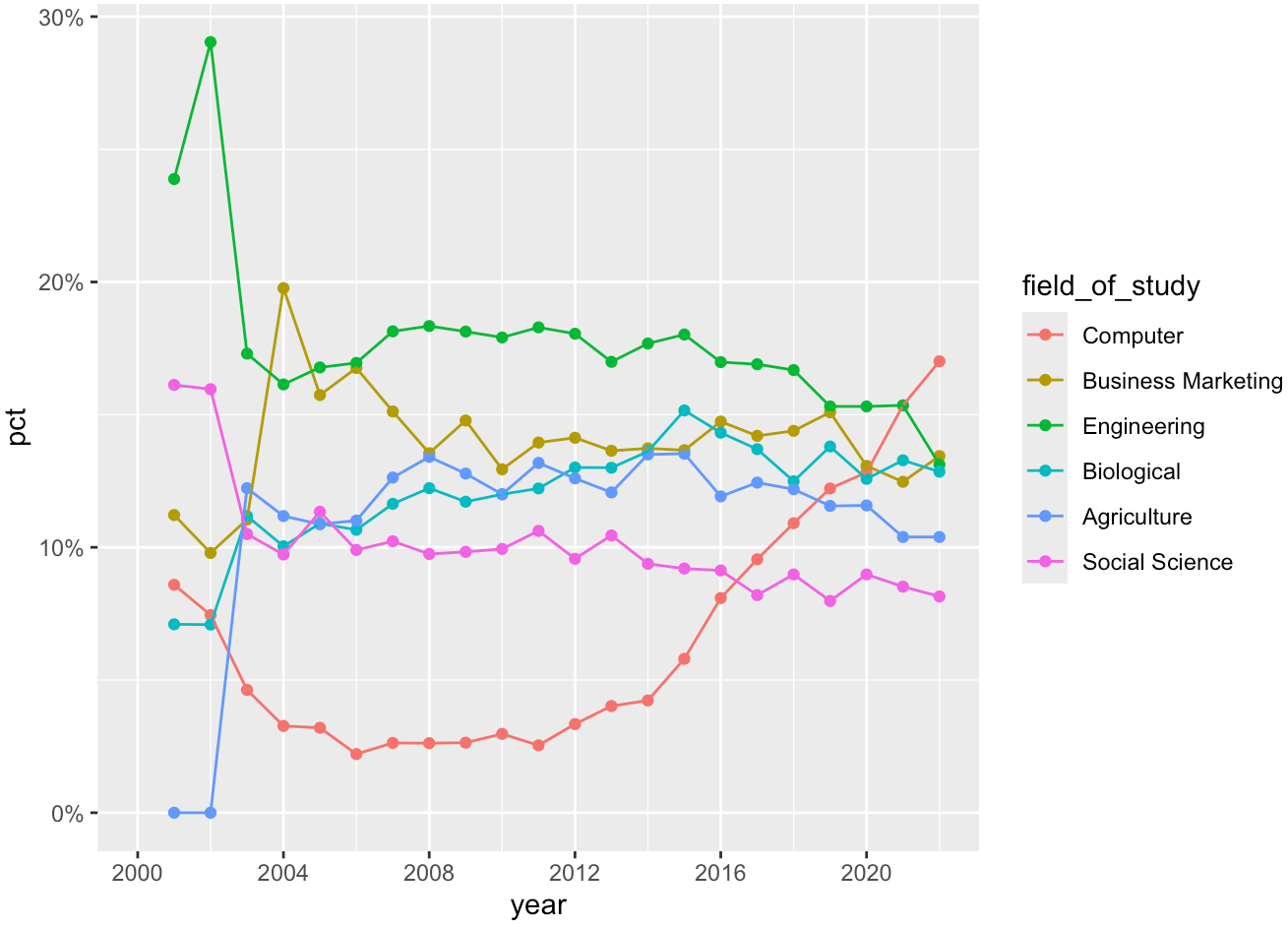
```
    values_to = "pct"
  ) |>
  ggplot(aes(x = year, y = pct, color = field_of_study)) +
  geom_point() +
  geom_line() +
  scale_x_continuous(limits = c(2000, 2022), breaks = seq(2000, 2020, 4)) +
  scale_y_continuous(labels = label_percent())
```



- **Demo:** Update the order of the values in the legend so they match the order of the lines in the plot. Do this by adding on to your pipeline from earlier.

```
cornell_deg |>
  pivot_longer(
    cols = -field_of_study,
    names_to = "year",
    names_transform = parse_number,
    values_to = "pct"
  ) |>
  mutate(
    field_of_study = fct_relevel(
      .f = field_of_study,
      "Computer", "Business Marketing", "Engineering",
      "Biological", "Agriculture", "Social Science"
    )
```

```
) |>
ggplot(aes(x = year, y = pct, color = field_of_study)) +
geom_point() +
geom_line() +
scale_x_continuous(limits = c(2000, 2022), breaks = seq(2000, 2020, 4)) +
scale_y_continuous(labels = label_percent())
```



Tip

Instead of coding the `field_of_study` values manually, you can use `fct_reorder2()` from the **forcats** package to reorder the levels of a factor based on the values of another variable.

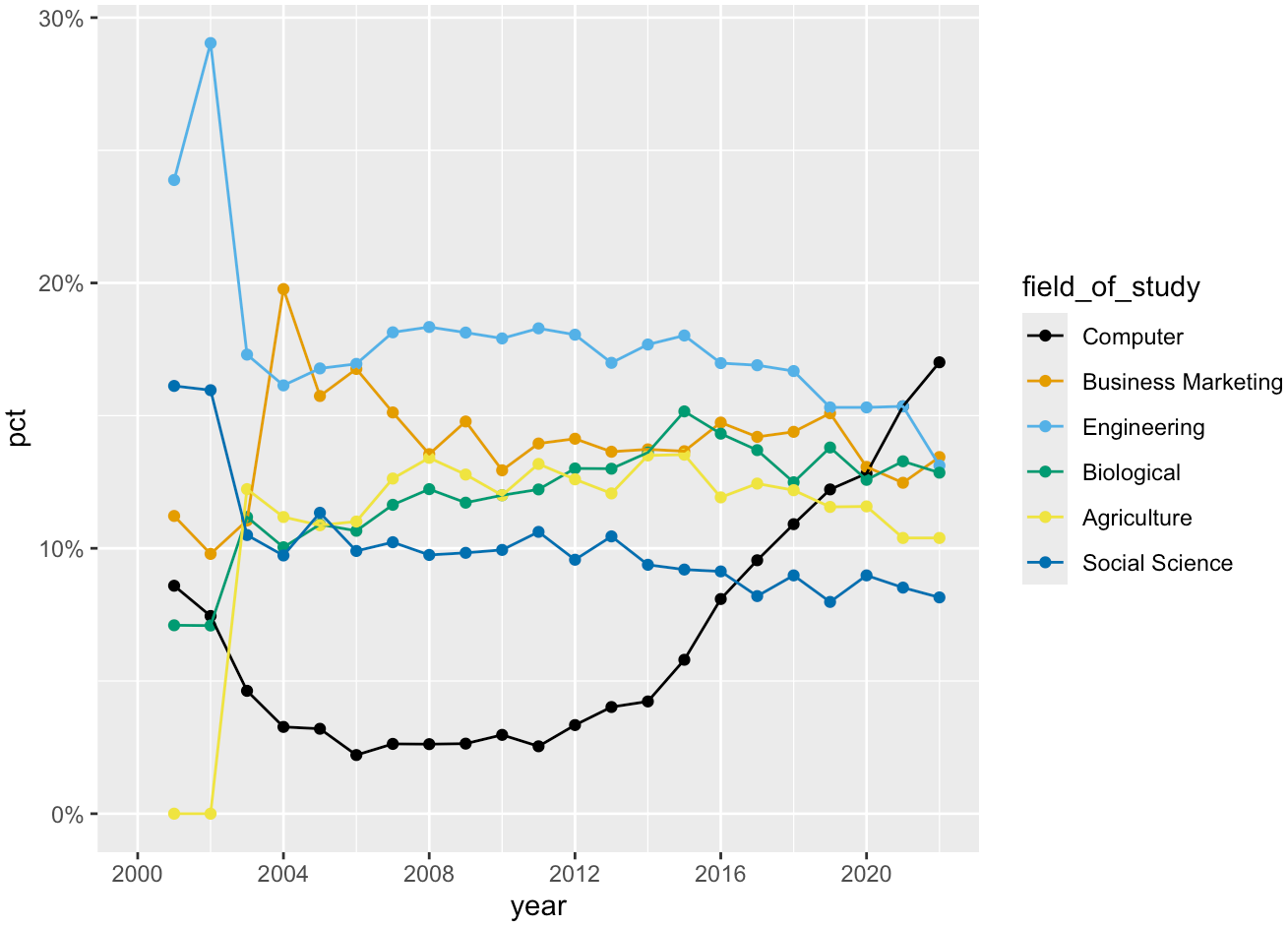
```
field_of_study = fct_reorder2(
  .f = field_of_study,
  .x = year,
  .y = pct
)
```

where it reorders the factor by the `.y` values associated with the largest `.x` values. This ensures the line colors in the legend match up to the end of the lines in the plot.

- **Demo:** Update line colors using the `scale_color_colorblind()` palette from **ggthemes**. Once again, do this by adding on to your pipeline from earlier.

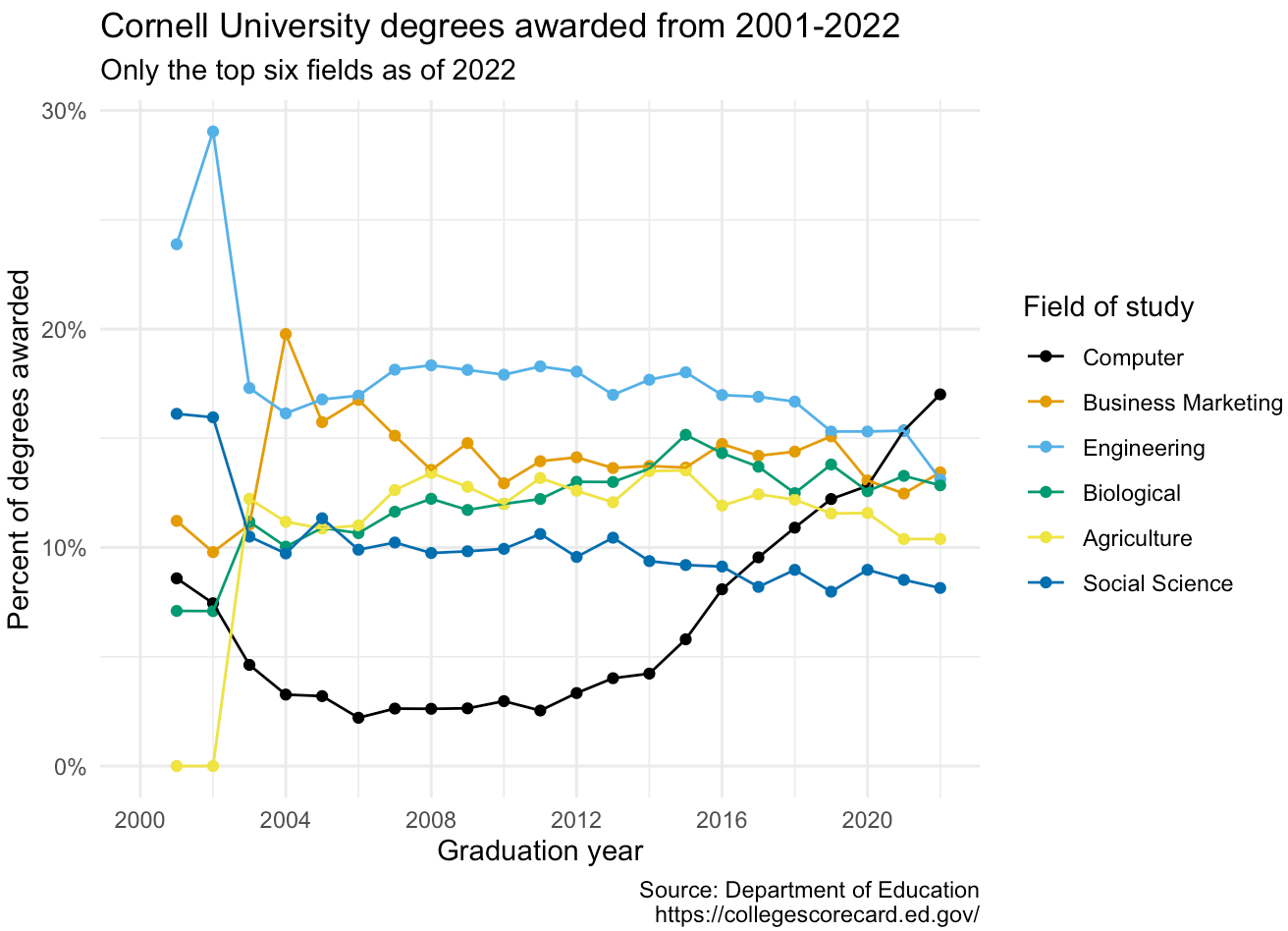
```
library(ggthemes)

cornell_deg |>
  pivot_longer(
    cols = -field_of_study,
    names_to = "year",
    names_transform = parse_number,
    values_to = "pct"
  ) |>
  mutate(
    field_of_study = fct_relevel(
      .f = field_of_study,
      "Computer", "Business Marketing", "Engineering",
      "Biological", "Agriculture", "Social Science"
    )
  ) |>
  ggplot(aes(x = year, y = pct, color = field_of_study)) +
  geom_point() +
  geom_line() +
  scale_x_continuous(limits = c(2000, 2022), breaks = seq(2000, 2020, 4)) +
  scale_y_continuous(labels = label_percent()) +
  scale_color_colorblind()
```



- **Your turn (4 minutes):** Update the plot labels (`title`, `subtitle`, `x`, `y`, and `caption`) and use `theme_minimal()`. Once again, do this by adding on to your pipeline from earlier.

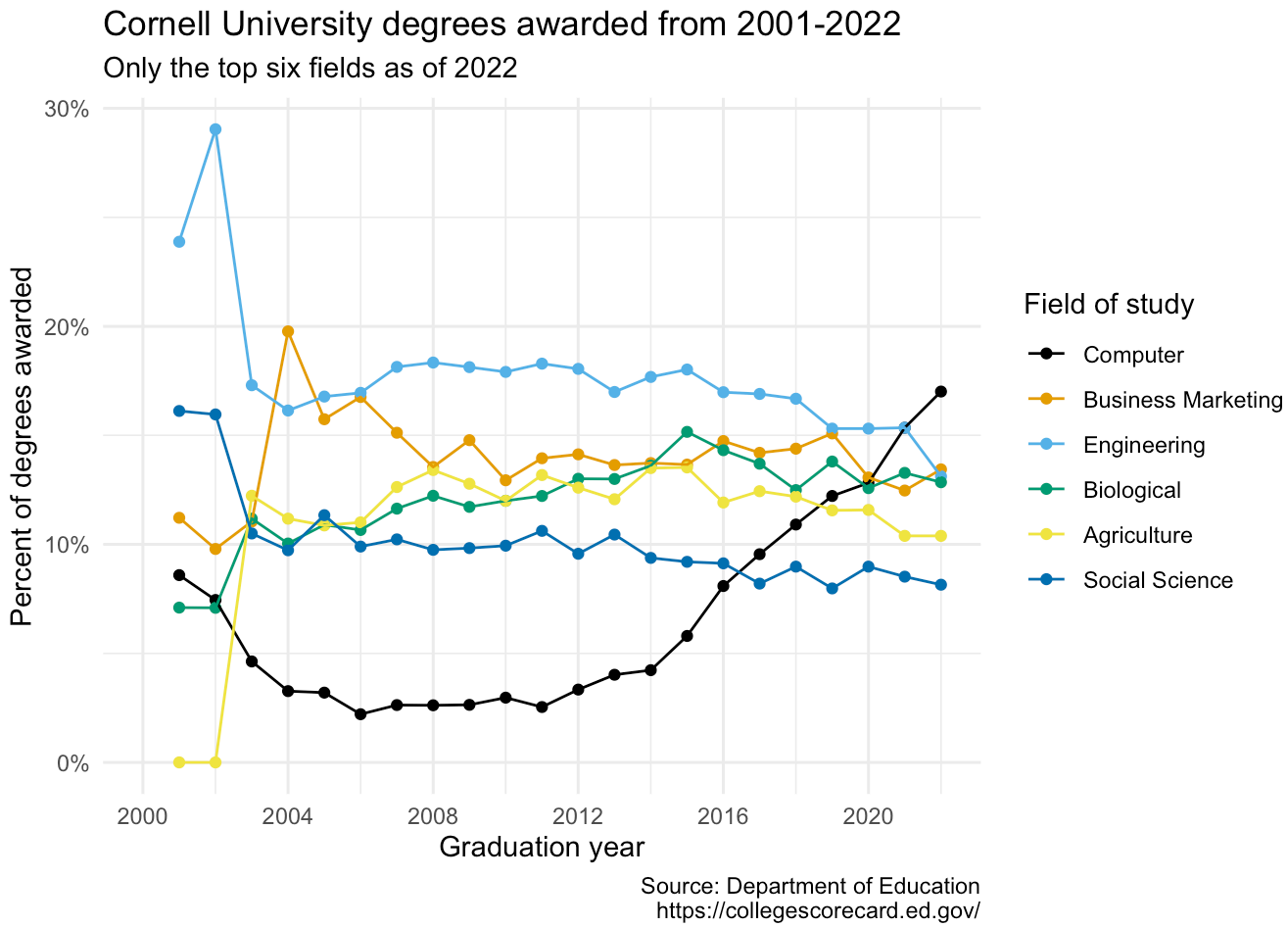
```
cornell_deg |>
  pivot_longer(
    cols = -field_of_study,
    names_to = "year",
    names_transform = parse_number,
    values_to = "pct"
  ) |>
  mutate(
    field_of_study = fct_relevel(
      .f = field_of_study,
      "Computer", "Business Marketing", "Engineering",
      "Biological", "Agriculture", "Social Science"
    )
  ) |>
  ggplot(aes(x = year, y = pct, color = field_of_study)) +
  geom_point() +
  geom_line() +
  scale_x_continuous(limits = c(2000, 2022), breaks = seq(2000, 2020, 4)) +
  scale_color_colorblind() +
  scale_y_continuous(labels = label_percent()) +
  labs(
    x = "Graduation year",
    y = "Percent of degrees awarded",
    color = "Field of study",
    title = "Cornell University degrees awarded from 2001-2022",
    subtitle = "Only the top six fields as of 2022",
    caption = "Source: Department of Education\nhttps://collegescorecard.ed.gov/"
  ) +
  theme_minimal()
```



- **Demo:** Finally, set `fig-width: 7` and `fig-height: 5` for your plot in the chunk options.

```
cornell_deg |>
  pivot_longer(
    cols = -field_of_study,
    names_to = "year",
    names_transform = parse_number,
    values_to = "pct"
  ) |>
  mutate(
    field_of_study = fct_relevel(
      .f = field_of_study,
      "Computer", "Business Marketing", "Engineering",
      "Biological", "Agriculture", "Social Science"
    )
  ) |>
  ggplot(aes(x = year, y = pct, color = field_of_study)) +
  geom_point() +
  geom_line() +
  scale_x_continuous(limits = c(2000, 2022), breaks = seq(2000, 2020, 4)) +
  scale_color_colorblind() +
  scale_y_continuous(labels = label_percent()) +
  labs(
    x = "Graduation year",
```

```
    y = "Percent of degrees awarded",
    color = "Field of study",
    title = "Cornell University degrees awarded from 2001-2022",
    subtitle = "Only the top six fields as of 2022",
    caption = "Source: Department of Education\nhttps://collegescorecard.ed.gov/"
  ) +
  theme_minimal()
```



Appendix: Alternative tidying strategy

Another tidying strategy suggested in class was to structure it one row for each year and one column for each of the fields of study. We could do this by transposing the data frame, which requires a `pivot_longer() |> pivot_wider()` approach:

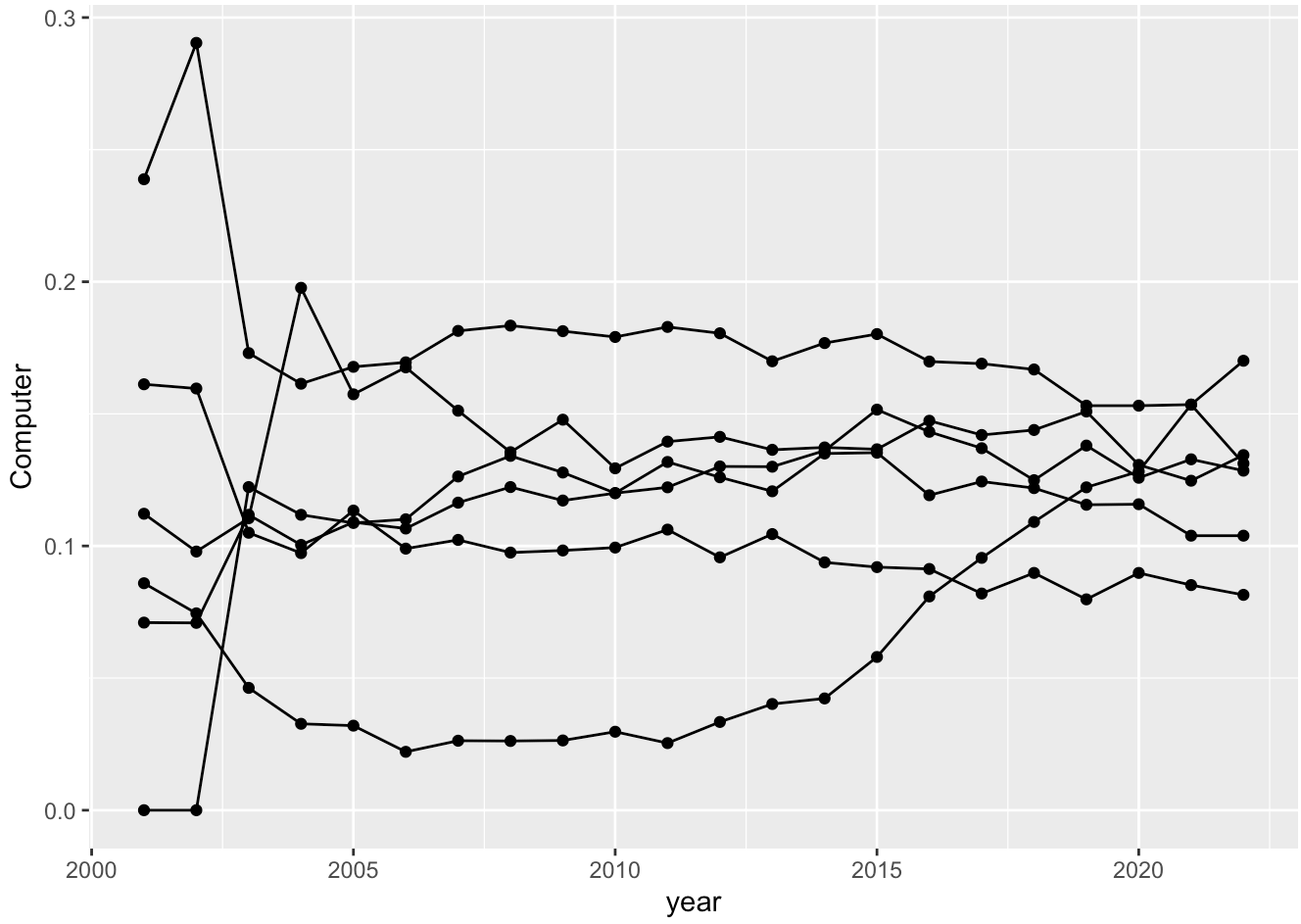
```
cornell_deg |>
  pivot_longer(
    cols = -field_of_study,
    names_to = "year",
    names_transform = parse_number,
    values_to = "pct"
  ) |>
  pivot_wider(
```

```
    names_from = field_of_study,
    values_from = pct
  )
```

```
# A tibble: 22 × 7
    year Computer `Business Marketing` Engineering Biological Agriculture
   <dbl>    <dbl>                <dbl>       <dbl>      <dbl>       <dbl>
 1  2001   0.0859               0.112        0.239     0.071        0
 2  2002   0.0745               0.0979       0.290     0.0709       0
 3  2003   0.0463               0.110        0.173     0.112        0.122
 4  2004   0.0327               0.198        0.161     0.100        0.112
 5  2005   0.032                0.157        0.168     0.109        0.109
 6  2006   0.0221               0.168        0.170     0.107        0.110
 7  2007   0.0263               0.151        0.181     0.116        0.126
 8  2008   0.0262               0.136        0.183     0.122        0.134
 9  2009   0.0264               0.148        0.181     0.117        0.128
10  2010   0.0297               0.129        0.179     0.12         0.12
# ℹ 12 more rows
# ℹ 1 more variable: `Social Science` <dbl>
```

But now we need to construct the line graph with the percentages spread across six columns. It would require us writing a separate `geom_*()` function for each field of study:

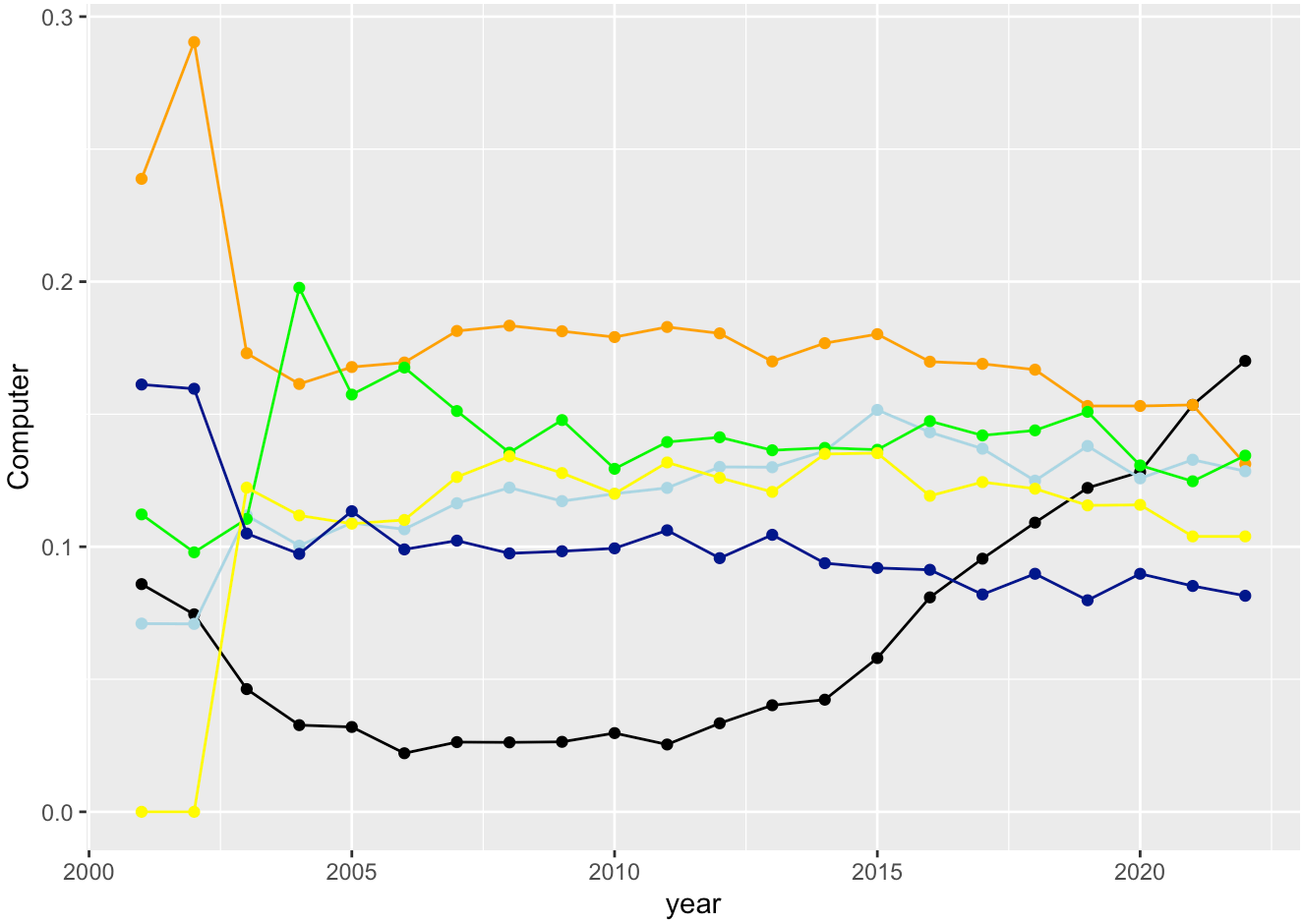
```
cornell_deg |>
  pivot_longer(
    cols = -field_of_study,
    names_to = "year",
    names_transform = parse_number,
    values_to = "pct"
  ) |>
  pivot_wider(
    names_from = field_of_study,
    values_from = pct
  ) |>
  ggplot(aes(x = year)) +
  geom_point(mapping = aes(y = Computer)) +
  geom_line(mapping = aes(y = Computer)) +
  geom_point(mapping = aes(y = Engineering)) +
  geom_line(mapping = aes(y = Engineering)) +
  geom_point(mapping = aes(y = Biological)) +
  geom_line(mapping = aes(y = Biological)) +
  geom_point(mapping = aes(y = `Business Marketing`)) +
  geom_line(mapping = aes(y = `Business Marketing`)) +
  geom_point(mapping = aes(y = Agriculture)) +
  geom_line(mapping = aes(y = Agriculture)) +
  geom_point(mapping = aes(y = `Social Science`)) +
  geom_line(mapping = aes(y = `Social Science`))
```



And we still don't have color-coding. We could use the `color` argument in each `geom_*()` function to change the color of each layer.

```
cornell_deg |>
  pivot_longer(
    cols = -field_of_study,
    names_to = "year",
    names_transform = parse_number,
    values_to = "pct"
  ) |>
  pivot_wider(
    names_from = field_of_study,
    values_from = pct
  ) |>
  ggplot(aes(x = year)) +
  geom_point(mapping = aes(y = Computer), color = "black") +
  geom_line(mapping = aes(y = Computer), color = "black") +
  geom_point(mapping = aes(y = Engineering), color = "orange") +
  geom_line(mapping = aes(y = Engineering), color = "orange") +
  geom_point(mapping = aes(y = Biological), color = "lightblue") +
  geom_line(mapping = aes(y = Biological), color = "lightblue") +
  geom_point(mapping = aes(y = `Business Marketing`), color = "green") +
  geom_line(mapping = aes(y = `Business Marketing`), color = "green") +
  geom_point(mapping = aes(y = Agriculture), color = "yellow") +
```

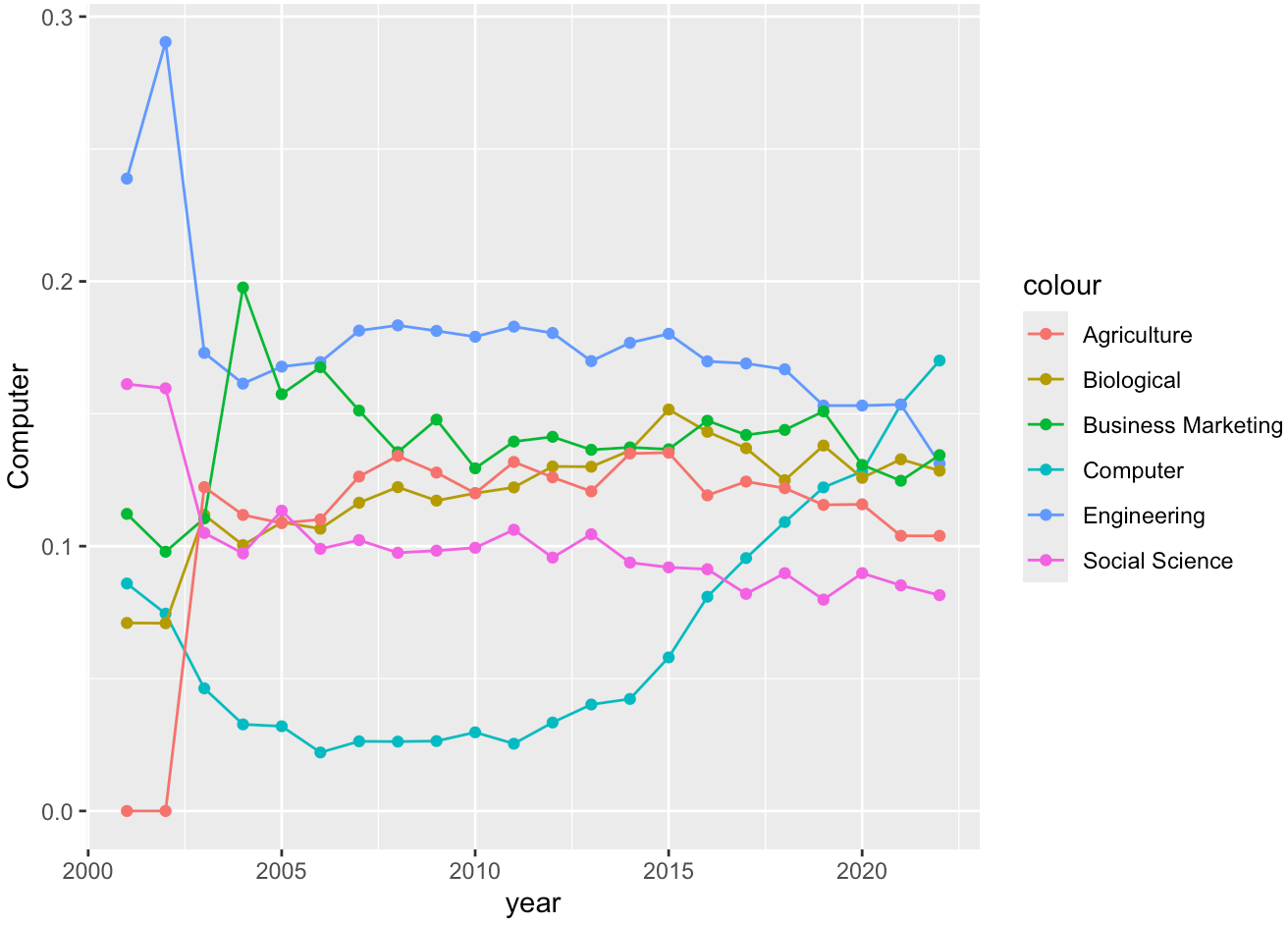
```
  geom_line(mapping = aes(y = Agriculture), color = "yellow") +
  geom_point(mapping = aes(y = `Social Science`), color = "darkblue") +
  geom_line(mapping = aes(y = `Social Science`), color = "darkblue")
```



But we still do not have a legend that tells us what each color represents. We want a legend generated automatically and that only happens if we map something to the `color` channel using `aes()`. We can hack this a bit by passing a character string within `aes()` to define a different unique value for each layer.

```
cornell_deg |>
  pivot_longer(
    cols = -field_of_study,
    names_to = "year",
    names_transform = parse_number,
    values_to = "pct"
  ) |>
  pivot_wider(
    names_from = field_of_study,
    values_from = pct
  ) |>
  ggplot(aes(x = year)) +
  geom_point(mapping = aes(y = Computer, color = "Computer")) +
  geom_line(mapping = aes(y = Computer, color = "Computer")) +
  geom_point(mapping = aes(y = Engineering, color = "Engineering")) +
  geom_line(mapping = aes(y = Engineering, color = "Engineering")) +
```

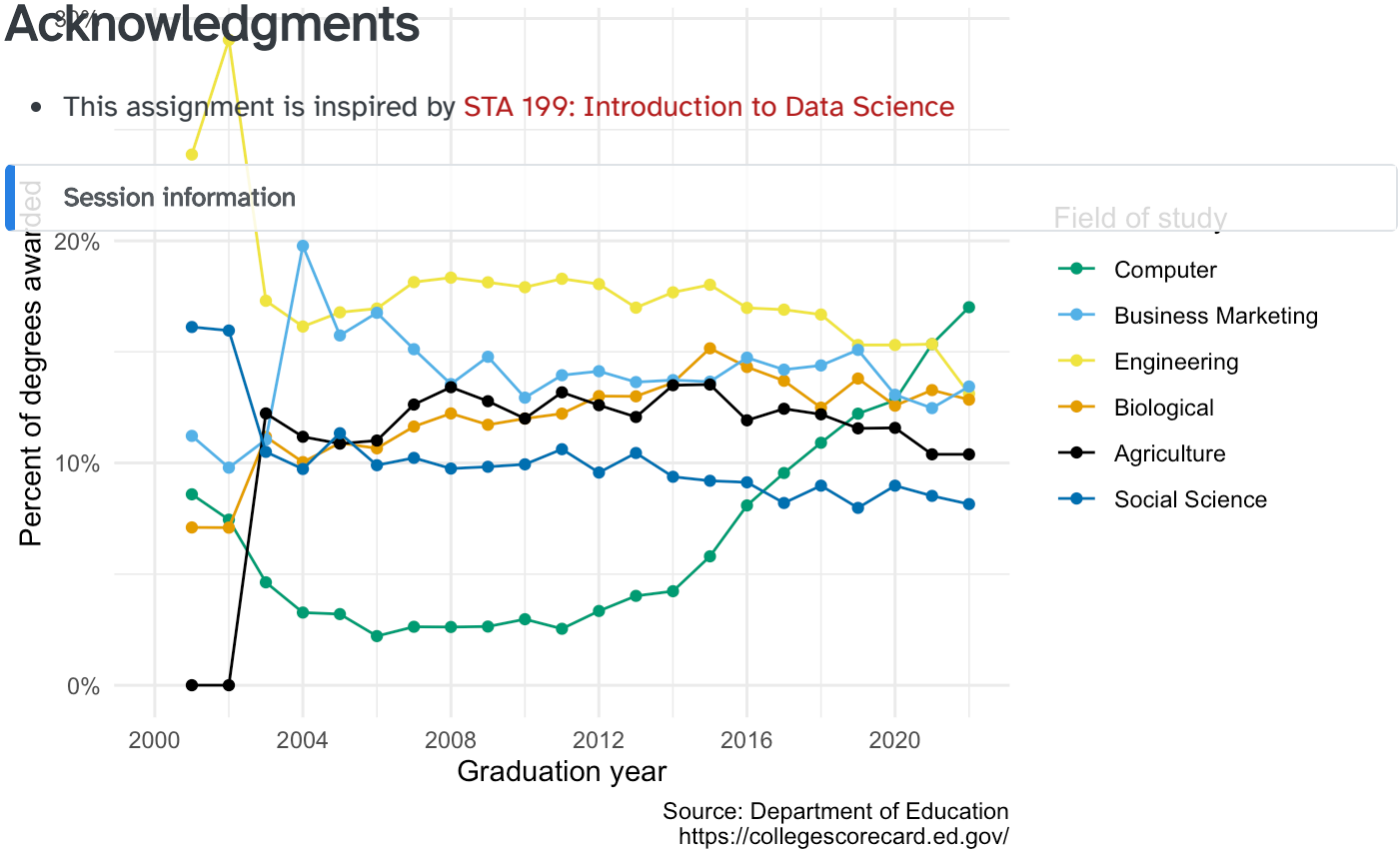
```
  geom_point(mapping = aes(y = Biological, color = "Biological")) +
  geom_line(mapping = aes(y = Biological, color = "Biological")) +
  geom_point(mapping = aes(y = `Business Marketing`, color = "Business Marketing")) +
  geom_line(mapping = aes(y = `Business Marketing`, color = "Business Marketing")) +
  geom_point(mapping = aes(y = Agriculture, color = "Agriculture")) +
  geom_line(mapping = aes(y = Agriculture, color = "Agriculture")) +
  geom_point(mapping = aes(y = `Social Science`, color = "Social Science")) +
  geom_line(mapping = aes(y = `Social Science`, color = "Social Science"))
```



Polished up we get the same plot.

```
cornell_deg |>
  pivot_longer(
    cols = -field_of_study,
    names_to = "year",
    names_transform = parse_number,
    values_to = "pct"
  ) |>
  pivot_wider(
    names_from = field_of_study,
    values_from = pct
  ) |>
  ggplot(aes(x = year)) +
  geom_point(mapping = aes(y = Computer, color = "Computer")) +
```

```
  geom_line(mapping = aes(y = Computer, color = "Computer")) +
  geom_point(mapping = aes(y = Engineering, color = "Engineering")) +
  geom_line(mapping = aes(y = Engineering, color = "Engineering")) +
  geom_point(mapping = aes(y = Biological, color = "Biological")) +
  geom_line(mapping = aes(y = Biological, color = "Biological")) +
  geom_point(mapping = aes(y = `Business Marketing`, color = "Business Marketing")) +
  geom_line(mapping = aes(y = `Business Marketing`, color = "Business Marketing")) +
  geom_point(mapping = aes(y = Agriculture, color = "Agriculture")) +
  geom_line(mapping = aes(y = Agriculture, color = "Agriculture")) +
  geom_point(mapping = aes(y = `Social Science`, color = "Social Science")) +
  geom_line(mapping = aes(y = `Social Science`, color = "Social Science")) +
  scale_x_continuous(limits = c(2000, 2022), breaks = seq(2000, 2020, 4)) +
  scale_color_colorblind(breaks = c(
    "Computer", "Business Marketing", "Engineering",
    "Biological", "Agriculture", "Social Science"
  )) +
  scale_y_continuous(labels = label_percent()) +
  labs(
    x = "Graduation year",
    y = "Percent of degrees awarded",
    color = "Field of study",
    title = "Cornell University degrees awarded from 2001-2022",
    subtitle = "Only the top six fields as of 2022",
    caption = "Source: Department of Education\nhttps://collegescorecard.ed.gov/"
  ) +
  theme_minimal()
```



Acknowledgments

- This assignment is inspired by STA 199: Introduction to Data Science

Session information

But with a lot more effort.