hw6

Cory Costello

November 7, 2017

1 Read and Tidy Data

This was a little tricky, since the spreadsheet had 7 lines at the beginning that needed to be skipped and a line at the end also need to be skipped.

I got through this by using the `skip = 7` argument to skip the first 7 lines, then you had to set `header = TRUE` to get the header in correctly (which started on line 8), then I passed that file to `slice()` to save just the first three rows (eliminating the 3 empty rows, and the final row of notes about the data).

Finally, I gathered the data (excluding the labor status variable), so that year was represented row-wise. Then, I had to use a combination of separates and unites to get percentage and se variables together, and ended by mutating to turn the percentage, se, and year into numeric variables (year required `parse_number()`, since each had an x in front of the year, as it was imported as column names).

Finally, I used bind_rows to combine the data, used `.id = TRUE` to maintain a record of the source dataset, and then changed those id's (1, 2, 3) to substantive labels (not_happy, pretty_happy, very_happy)

```r
library(tidyverse)
```

```
## Loading tidyverse: ggplot2
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr

## Conflicts with tidy packages ----------------------------------------------

## filter(): dplyr, stats
## lag():    dplyr, stats
```

```r
library(rio)

not_happy <- import("not_happy.csv", skip = 7, header = TRUE) %>%
  janitor::clean_names() %>%
  slice(1:3) %>%
  gather(year, pct_endorsement, -labor_force_status) %>%
  separate(pct_endorsement, c("pct_endorsement", "pct_dec",
                              "se_endorsement", "se_dec")) %>%
  unite(pct_endorsement, pct_endorsement:pct_dec, sep = ".") %>%
  unite(se_endorsement, se_endorsement:se_dec, sep = ".") %>%
  mutate(pct_endorsement = as.numeric(pct_endorsement),
         se_endorsement = as.numeric(se_endorsement),
         year = parse_number(year))
```

```
## Warning: Too many values at 93 locations: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
## 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...
```

```r
pretty_happy <- import("pretty_happy.csv", skip = 7, header = TRUE) %>%
  janitor::clean_names()%>%
  slice(1:3) %>%
  gather(year, pct_endorsement, -labor_force_status) %>%
  separate(pct_endorsement, c("pct_endorsement", "pct_dec",
                             "se_endorsement", "se_dec")) %>%
  unite(pct_endorsement, pct_endorsement:pct_dec, sep = ".") %>%
  unite(se_endorsement, se_endorsement:se_dec, sep = ".") %>%
  mutate(pct_endorsement = as.numeric(pct_endorsement),
         se_endorsement = as.numeric(se_endorsement),
         year = parse_number(year))
```

```
## Warning: Too many values at 93 locations: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
## 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...
```

```
very_happy <- import("very_happy.csv", skip = 7, header = TRUE) %>%
  janitor::clean_names()%>%
  slice(1:3) %>%
  gather(year, pct_endorsement, -labor_force_status) %>%
  separate(pct_endorsement, c("pct_endorsement", "pct_dec",
                             "se_endorsement", "se_dec")) %>%
  unite(pct_endorsement, pct_endorsement:pct_dec, sep = ".") %>%
  unite(se_endorsement, se_endorsement:se_dec, sep = ".") %>%
  mutate(pct_endorsement = as.numeric(pct_endorsement),
         se_endorsement = as.numeric(se_endorsement),
         year = parse_number(year))
```

```
## Warning: Too many values at 93 locations: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
## 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...
```

```
full_data <- bind_rows(not_happy, pretty_happy, very_happy, .id = "happiness") %>%
  mutate(happiness = ifelse(happiness == 1, "not_happy",
                          ifelse(happiness == 2, "pretty_happy", "very_happy")))
```

2 Answering some questions

```
highest_pct_vhappy<- full_data %>%
  filter(happiness == "very_happy") %>%
  arrange(desc(pct_endorsement)) %>%
  slice(1)

highest_pct_vhappy
```

```
## # A tibble: 1 x 5
##    happiness       labor_force_status  year pct_endorsement se_endorsement
##        <chr>                    <chr> <dbl>           <dbl>          <dbl>
## 1 very_happy Not in labor force/Other  1984              42           2.81
```

It looks like the highest percentage of folks that endorsed being very happy were Not in labor force/Other in the year 1984; 42% of them endorsed being very happy.

```
pct_happiness_employed<- full_data %>%
  filter(labor_force_status == "Employed") %>%
  group_by(happiness) %>%
  summarize(m_pct_endorsement = mean(pct_endorsement, na.rm = TRUE),
            m_se_endorsement  = mean(se_endorsement, na.rm = TRUE))
```

```
pct_happiness_employed
```

```
## # A tibble: 3 x 3
##      happiness m_pct_endorsement m_se_endorsement
##          <chr>             <dbl>            <dbl>
## 1    not_happy          9.687097        0.8764516
## 2 pretty_happy         57.193548        1.5003226
## 3   very_happy         33.077419        1.4551613
```

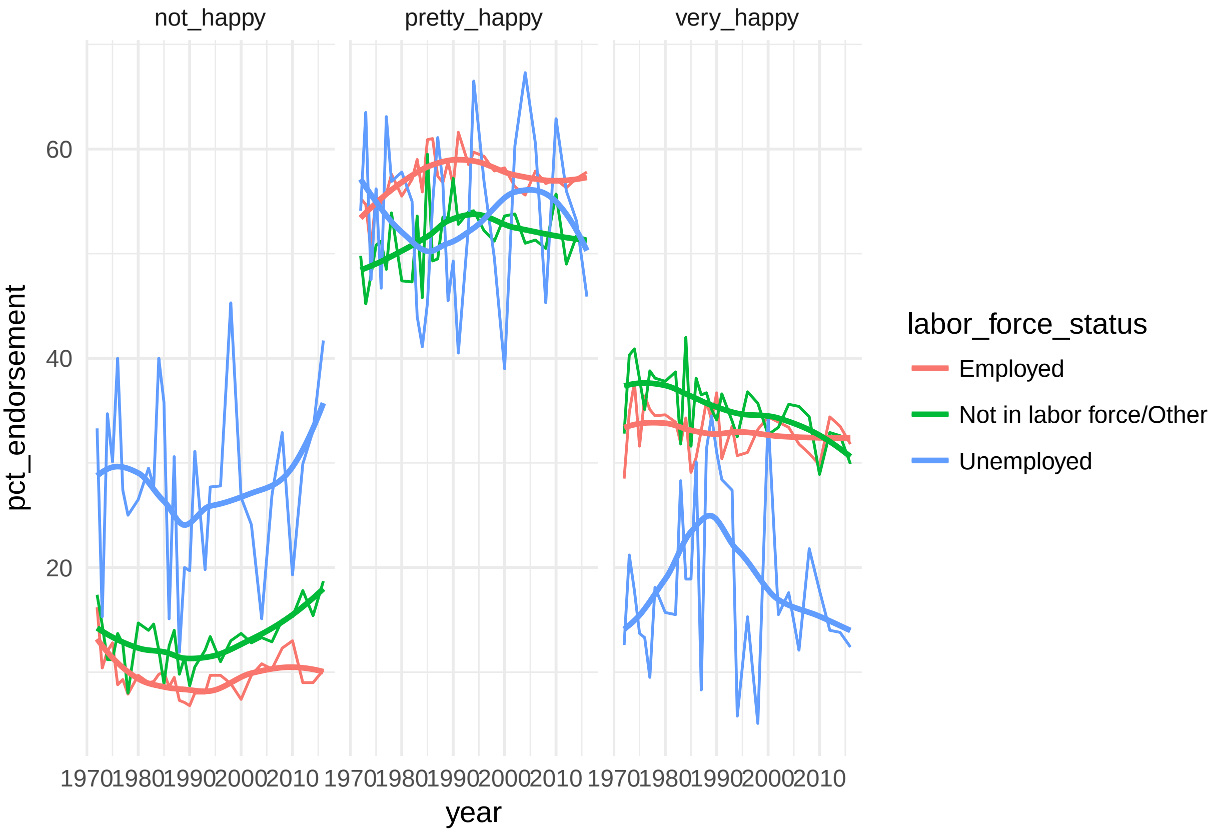
It looks like, within respondents whose labor force satus was reported as Employed, 9.69% indicated that they were “not happy”, 57.19% indicated that they were “pretty happy”, and 33.08% indicated that they were “very happy”, averaged across available years.

3 Plots!!

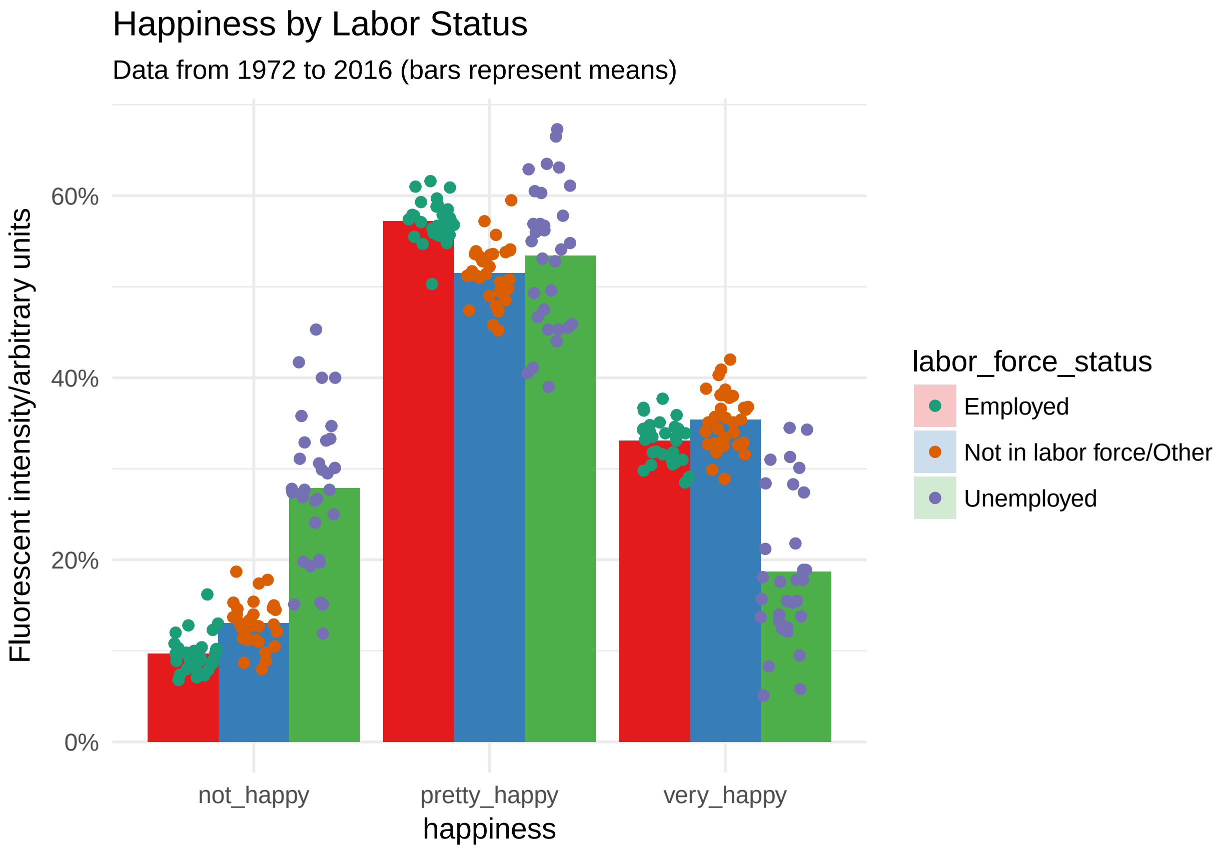
```
library(ggthemes)
theme_set(theme_minimal())

ggplot(full_data, aes(x = year, y = pct_endorsement, color = labor_force_status)) +
  geom_line()+
  geom_smooth(se= FALSE)+
  facet_wrap(~happiness, nrow = 1)
```

```
## `geom_smooth()` using method = 'loess'
```



```
full_data %>%
  group_by (labor_force_status, happiness) %>%
  mutate(m_pct_endorsement = (mean(pct_endorsement))/100,
         pct_endorsement = pct_endorsement/100) %>%
  ggplot(aes(x = happiness, y = pct_endorsement))+
  geom_bar(aes(y = m_pct_endorsement, fill = labor_force_status),
           stat = "identity", position = "dodge", alpha = .25)+
  geom_point(aes(y = pct_endorsement, color = labor_force_status),
             position =  position_jitterdodge(jitter.width = .2))+
  scale_fill_brewer(palette = "Set1")+
  scale_colour_brewer(palette = "Dark2")+
  scale_y_continuous(name="Fluorescent intensity/arbitrary units", labels = scales::percent) +
  ggtitle("Happiness by Labor Status", subtitle = "Data from 1972 to 2016 (bars represent means)")
```



I just wanted to say somewhere (so I'm putting it here) that getting the percentage sign on the y-axis was pretty tough. I ended up transforming the variables to proportions (.xx) from percentage (xx.xx) and then using the `scales::percent()` function within the `scale_y_continuous()` function. I'm curious if there is an easier way (which I guess you'll probably cover tomorrow).