HW exemplar

```
#html_document: default
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.5.3
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.1
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## Warning: package 'ggplot2' was built under R version 3.5.3
```

```
## Warning: package 'tibble' was built under R version 3.5.3
```

```
## Warning: package 'tidyr' was built under R version 3.5.3
```

```
## Warning: package 'purrr' was built under R version 3.5.3
```

```
## Warning: package 'dplyr' was built under R version 3.5.3
```

```
## Warning: package 'stringr' was built under R version 3.5.3
```

```
## Warning: package 'forcats' was built under R version 3.5.3
```

```
## -- Conflicts ------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(forcats)
library(RColorBrewer)
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 3.5.3
```

```
library(tidytext)
```

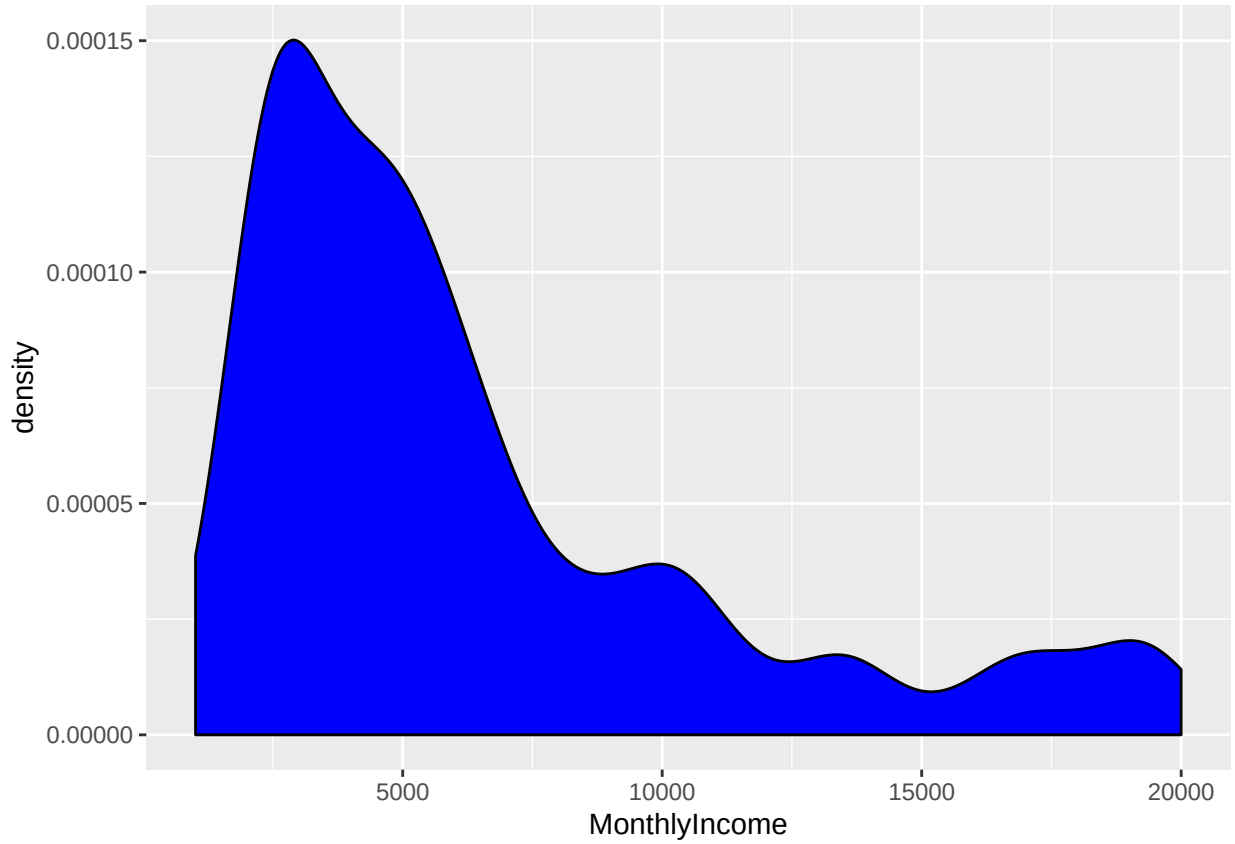
```
## Warning: package 'tidytext' was built under R version 3.5.3
```

Loading Data

```
load("C:/Users/jerem/Google Drive/Online/Vandy/llo8200repo/attrition.Rdata")
```

1. Create a graph that shows the distribution of monthly income.

```
gg<-ggplot(at,aes(x=MonthlyIncome))
gg<-gg+geom_density(fill="blue")
gg
```



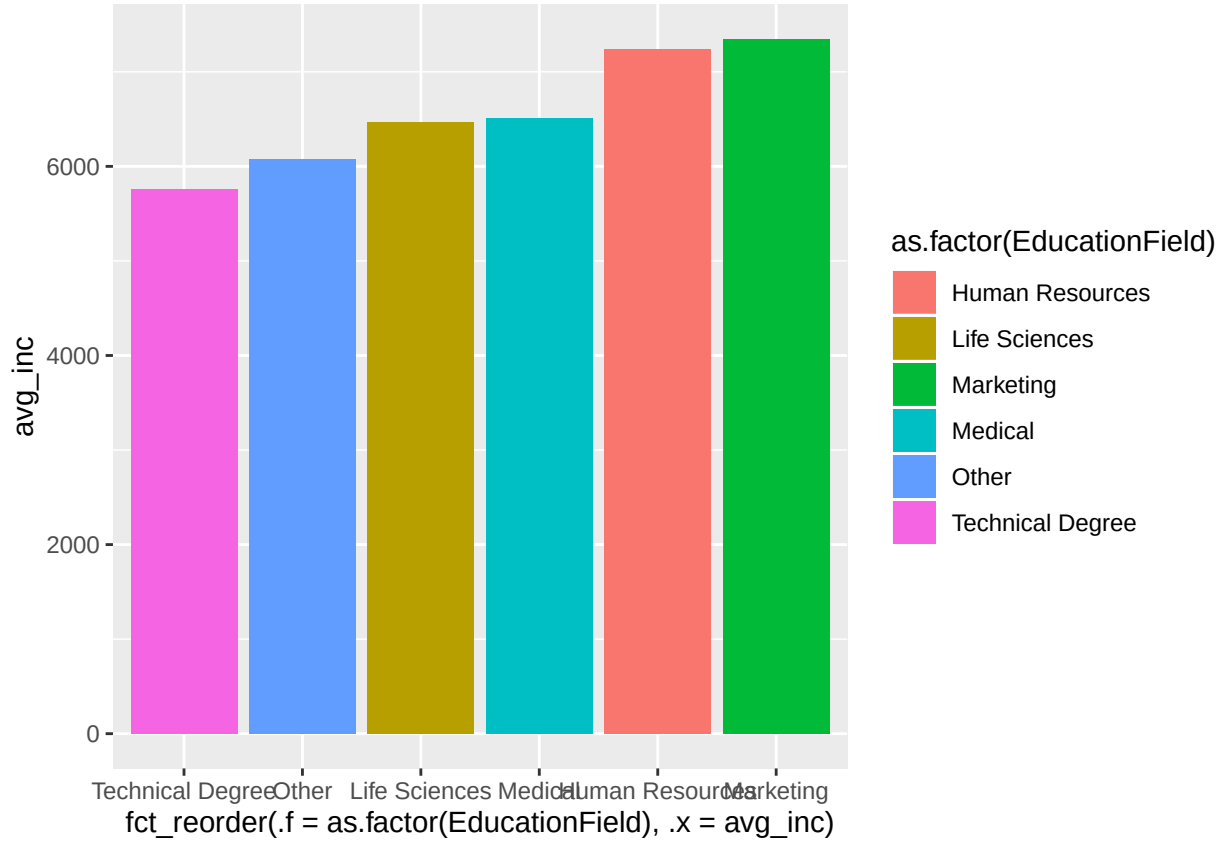
2. Create a graph that shows the average level of monthly income by field of education.

```
at_sum<-at%>%
  group_by(EducationField)%>%
  summarize(avg_inc=mean(MonthlyIncome))


at_sum
```

```
## # A tibble: 6 x 2
##   EducationField   avg_inc
##   <chr>              <dbl>
## 1 Human Resources    7241.
## 2 Life Sciences      6463.
## 3 Marketing          7349.
```

```
## 4 Medical            6510.
## 5 Other              6072.
## 6 Technical Degree   5758.
```

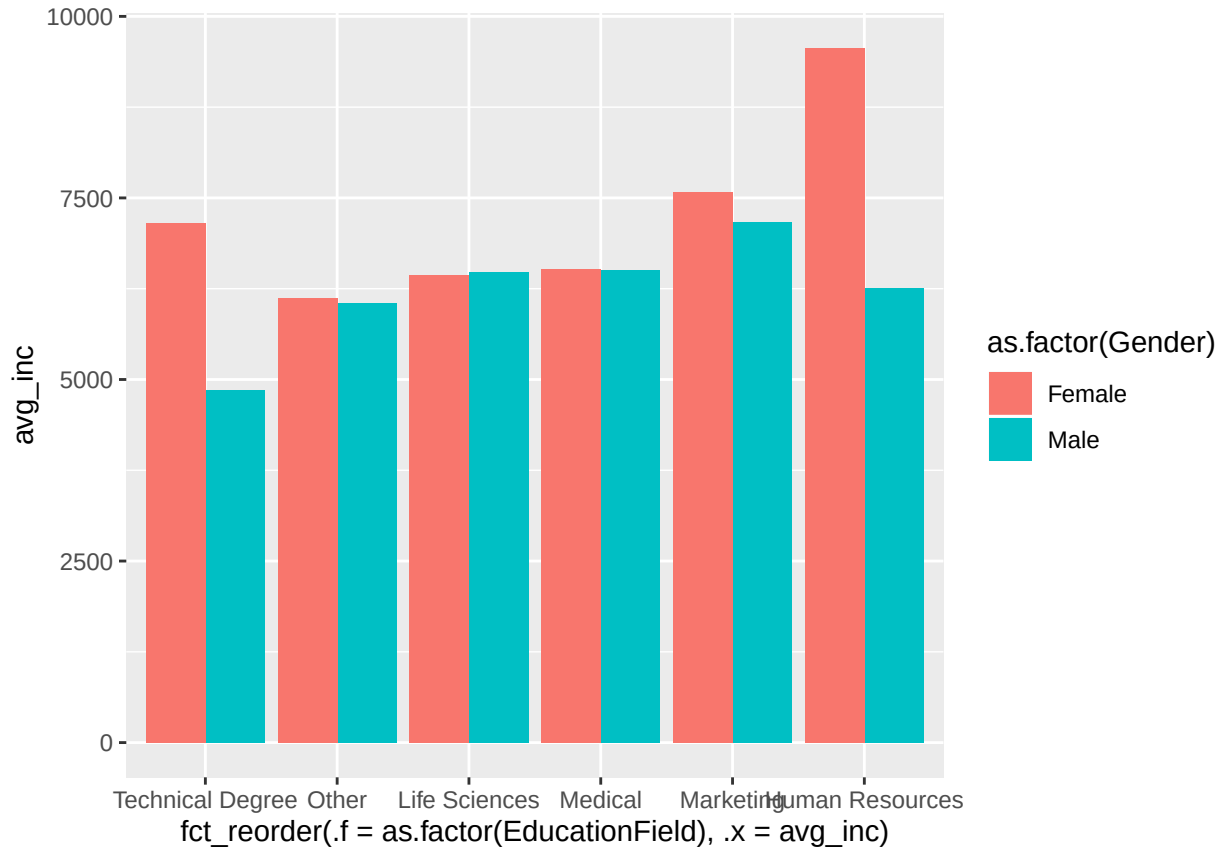
```
gg_education<-ggplot(at_sum,aes(x=fct_reorder(.f=as.factor(EducationField),
                                              .x=avg_inc),
                                y=avg_inc,
                                fill=as.factor(EducationField)))
## Use bar plot geometry, height of bars set by level observed in dataset
gg_education<-gg_education+geom_bar(stat="Identity")
## Print
gg_education
```



3. Create another graph that shows average level of monthly income by field of education and gender.

```
at_sum<-at%>%
  group_by(EducationField,Gender)%>%
  summarize(avg_inc=mean(MonthlyIncome))


gg<-ggplot(at_sum,aes(x=fct_reorder(.f=as.factor(EducationField),
                                              .x=avg_inc),
                                y=avg_inc,
                                fill=as.factor(Gender)))
## Use bar plot geometry, height of bars set by level observed in dataset
```

```
gg<-gg+geom_bar(stat="Identity",position="dodge")
## Print
gg
```

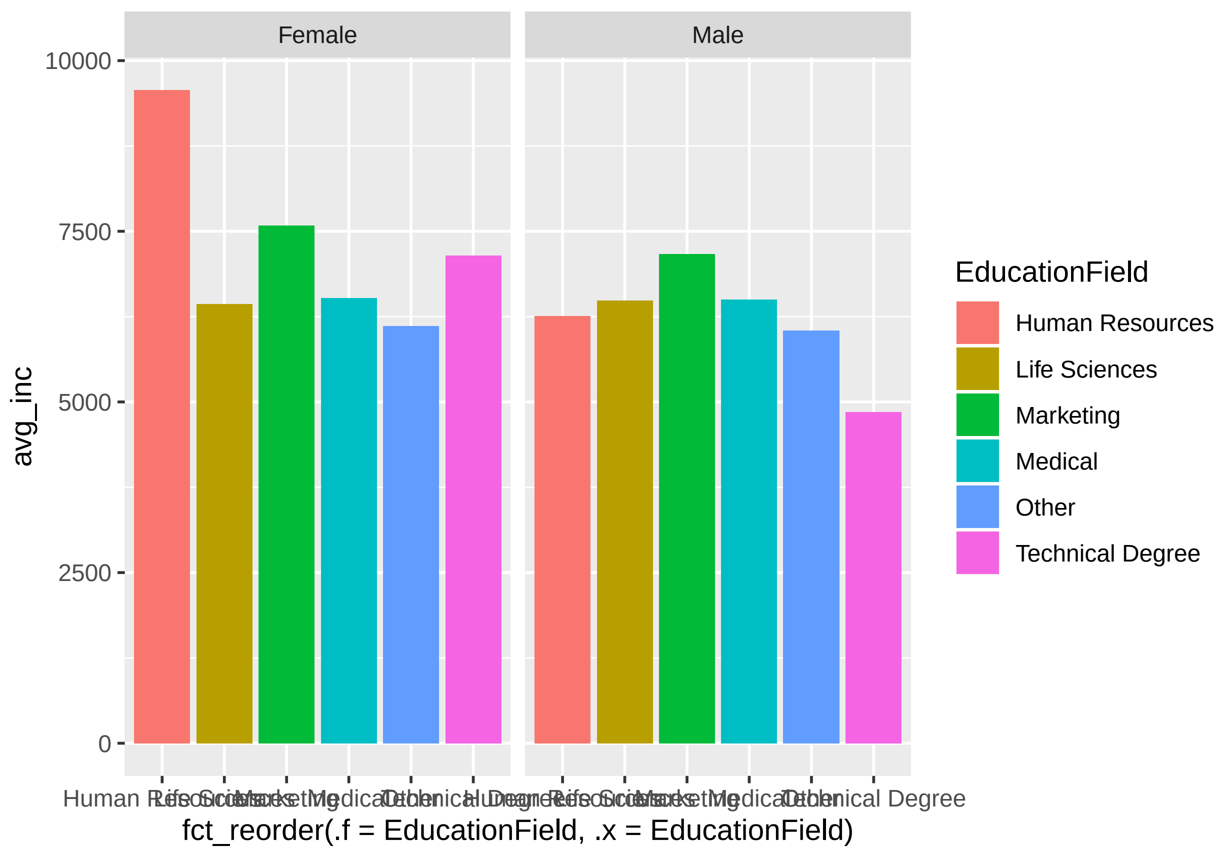


. . . . Or we could display this using facet wrap . . .

```
gg<-ggplot(at_sum,aes(x=fct_reorder(.f=EducationField,
                                    .x=EducationField),
                      y=avg_inc,
                      fill=EducationField))
## Use bar plot geometry, height of bars set by level observed in dataset
gg<-gg+geom_bar(stat="Identity",position="dodge")
gg<-gg+facet_wrap(~Gender)
## Print
gg
```

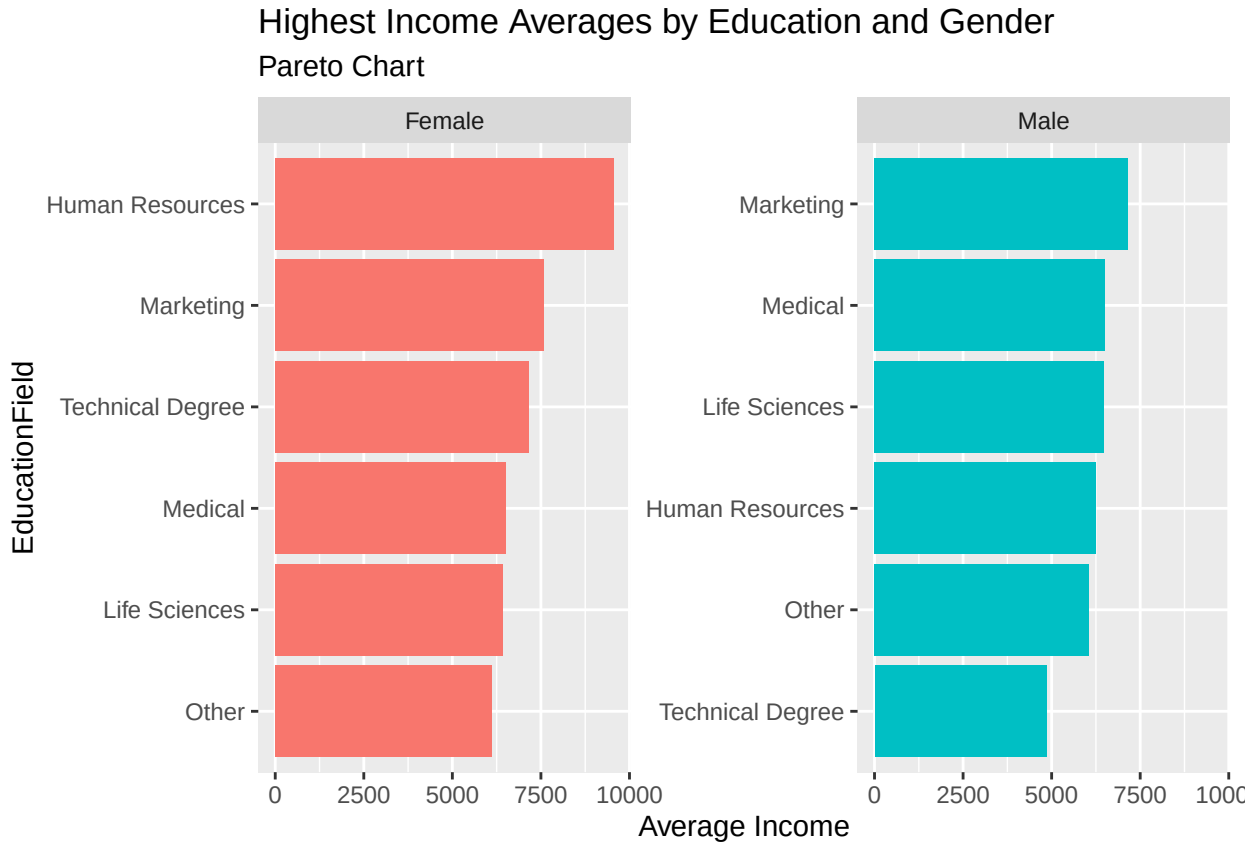
```
## Warning in mean.default(sort(x, partial = half + 0L:1L)[half + 0L:1L]): argument
## is not numeric or logical: returning NA

## Warning in mean.default(sort(x, partial = half + 0L:1L)[half + 0L:1L]): argument
## is not numeric or logical: returning NA

## Warning in mean.default(sort(x, partial = half + 0L:1L)[half + 0L:1L]): argument
## is not numeric or logical: returning NA
```

```
## Warning in mean.default(sort(x, partial = half + 0L:1L)[half + 0L:1L]): argument
## is not numeric or logical: returning NA

## Warning in mean.default(sort(x, partial = half + 0L:1L)[half + 0L:1L]): argument
## is not numeric or logical: returning NA

## Warning in mean.default(sort(x, partial = half + 0L:1L)[half + 0L:1L]): argument
## is not numeric or logical: returning NA

## Warning in mean.default(sort(x, partial = half + 0L:1L)[half + 0L:1L]): argument
## is not numeric or logical: returning NA

## Warning in mean.default(sort(x, partial = half + 0L:1L)[half + 0L:1L]): argument
## is not numeric or logical: returning NA

## Warning in mean.default(sort(x, partial = half + 0L:1L)[half + 0L:1L]): argument
## is not numeric or logical: returning NA

## Warning in mean.default(sort(x, partial = half + 0L:1L)[half + 0L:1L]): argument
## is not numeric or logical: returning NA

## Warning in mean.default(sort(x, partial = half + 0L:1L)[half + 0L:1L]): argument
## is not numeric or logical: returning NA

## Warning in mean.default(sort(x, partial = half + 0L:1L)[half + 0L:1L]): argument
## is not numeric or logical: returning NA
```



... well that doesn't look great ... AND the bars are not re-ordered?!?! Why not? As it happens, reorder does not work well in combination with facet wrap ... so lets use reorder_within.

```
at_sum %>%
    group_by(Gender) %>%
#     top_n(15) %>%
    ungroup %>%
    mutate(Gender = as.factor(Gender),
           EducationField = reorder_within(EducationField, avg_inc, Gender)) %>%
    ggplot(aes(EducationField, avg_inc, fill = Gender)) +
    geom_col(show.legend = FALSE) +
    facet_wrap(~Gender, scales = "free_y") +
  coord_flip() +
    scale_x_reordered() +
    labs(y = "Average Income",
         title = "Highest Income Averages by Education and Gender",
         subtitle = "Pareto Chart")
```

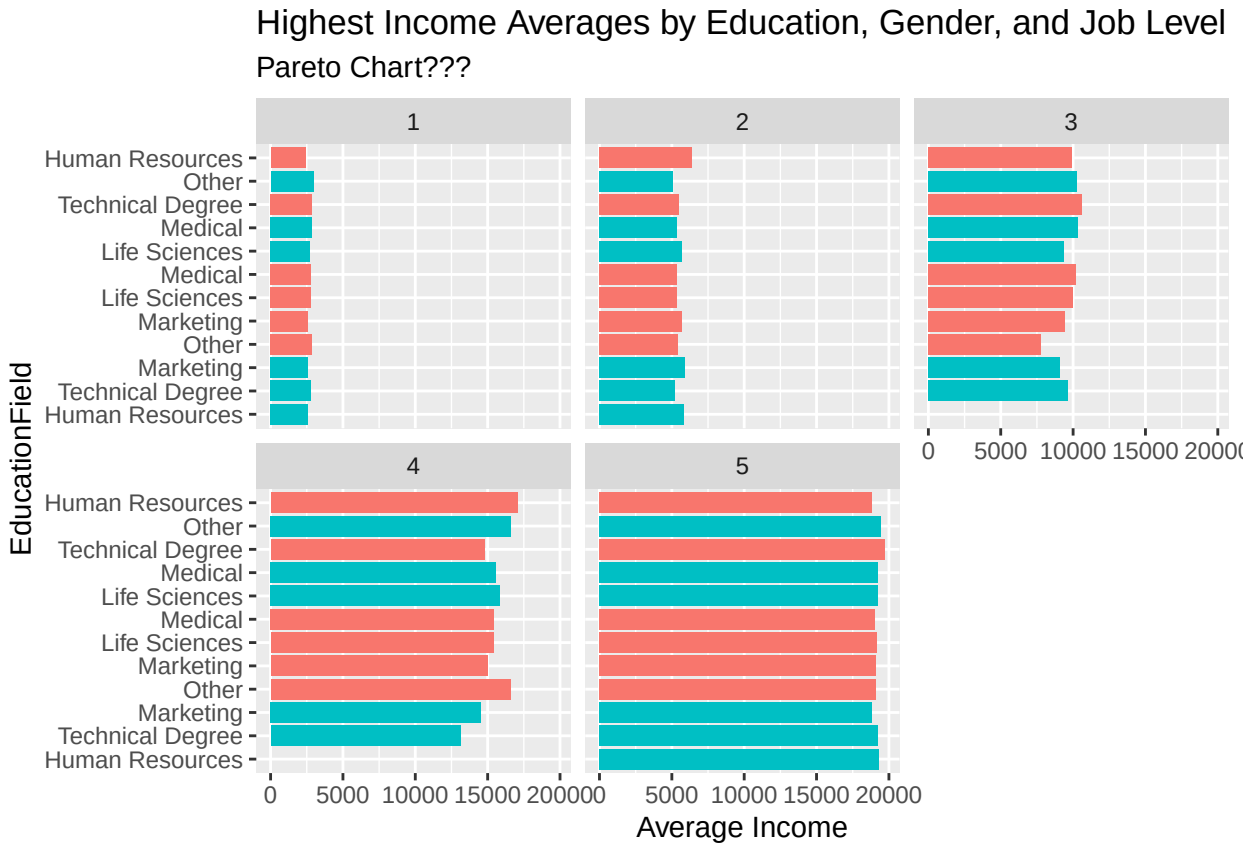


Great!! Each subfigure is a Paretto Chart (the bars are reordered!!!)

4. Create a graph that shows average levels of monthly income by field of education, gender and job level (scale of 1-5, highest ranked employees are

5)

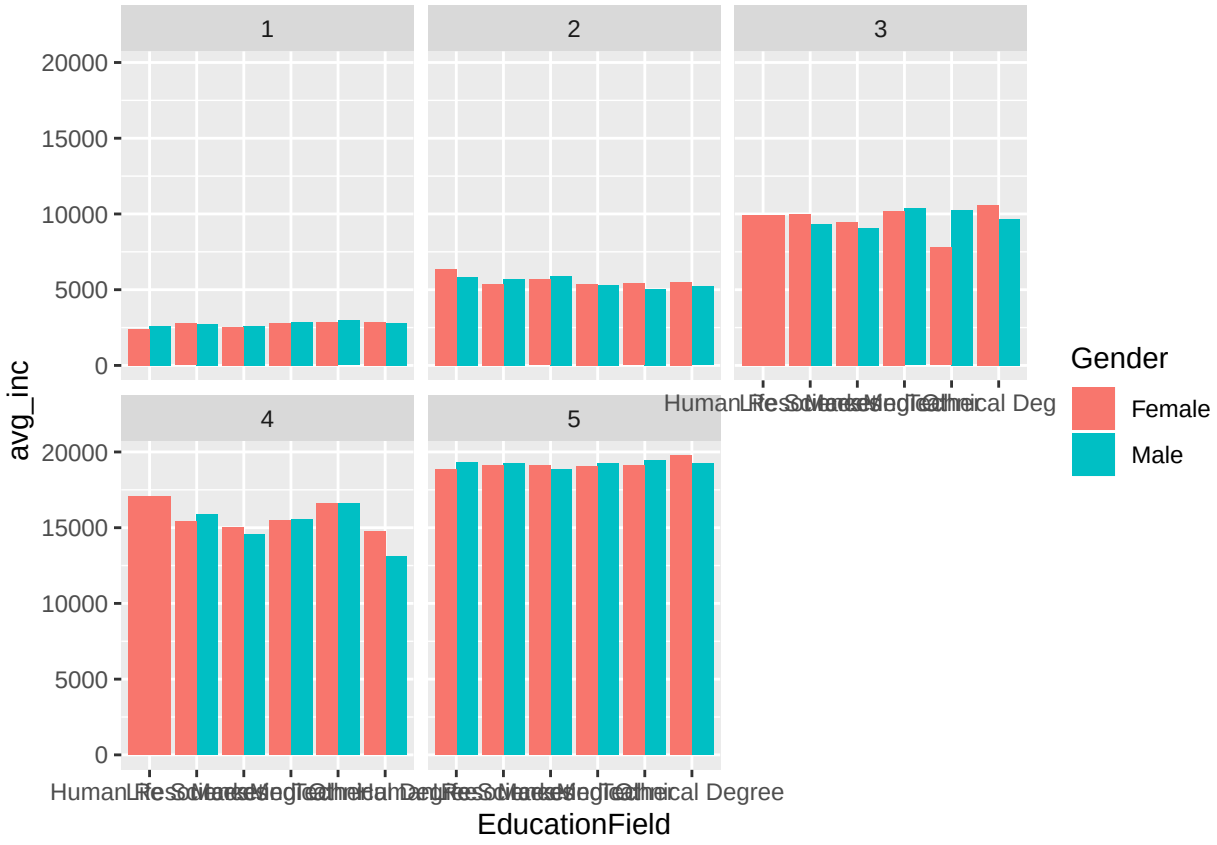
```
at_sum<-at%>%
  group_by(EducationField,Gender,JobLevel)%>%
  summarize(avg_inc=mean(MonthlyIncome))

at_sum %>%
    group_by(EducationField, Gender, JobLevel) %>%
#     top_n(6) %>%
    ungroup %>%
    mutate(EducationField = reorder_within(EducationField, avg_inc, Gender)) %>%
    ggplot(aes(EducationField, avg_inc, fill = Gender), position = "fill") +
    geom_col(show.legend = FALSE) +
    facet_wrap(~JobLevel) +
    scale_x_reordered() +
  coord_flip() +
    labs(y = "Average Income",
         title = "Highest Income Averages by Education, Gender, and Job Level ",
         subtitle = "Pareto Chart???")
```



Note: reorder_within is unintuitive when we have multiple bars per group!!! No Pareto here ... lets try someother plots for fun!!!

```
at_sum<-at%>%
  group_by(EducationField,Gender,JobLevel)%>%
  summarize(avg_inc=mean(MonthlyIncome))
```

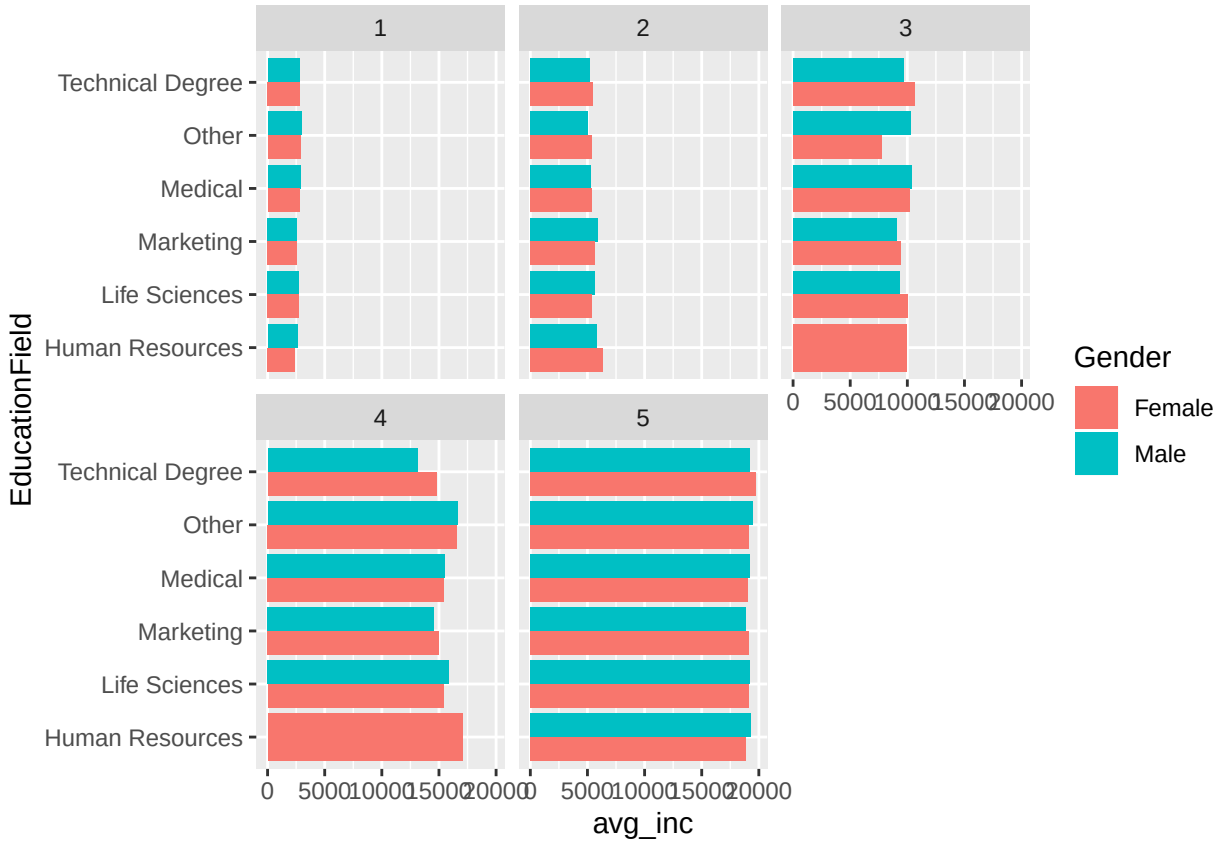
```
gg<-ggplot(at_sum,aes(x=EducationField,
                      y=avg_inc,
                      fill=Gender))
## Use bar plot geometry, height of bars set by level observed in dataset
gg<-gg+geom_bar(stat="Identity",position="dodge")
## Print
gg<-gg+facet_wrap(~JobLevel)
gg
```



well ... this figure can certainly be improved. Lets try to arrange the x labels so that they are actually readable.

We can use coord_flip ... as we have seen previously ....

```
gg <- gg + coord_flip()
gg
```

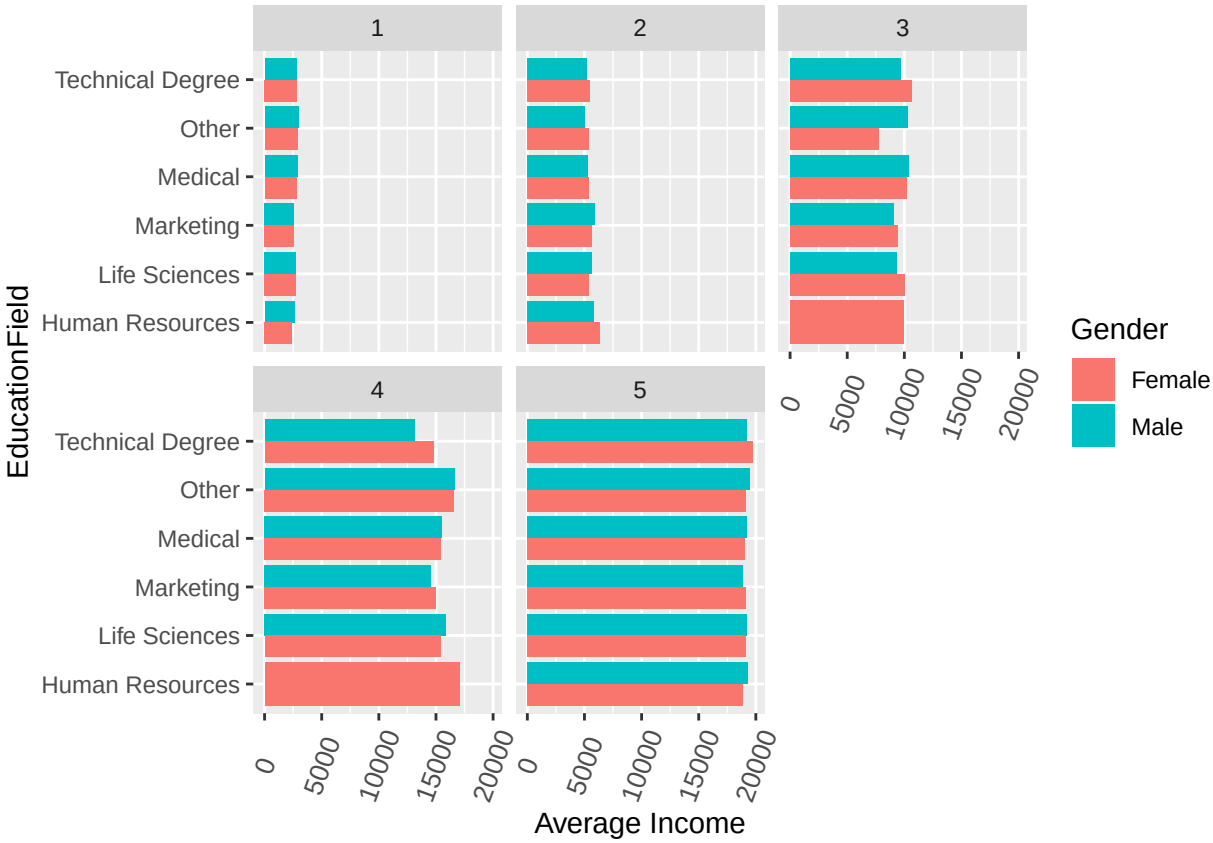


Hmmmm . . . still not great!!!

GGPLOT is based on the grammar of graphics . . . lots of details to cover (two entire texts – YIKES). Here is a brief tutorial: http://r-statistics.co/Complete-Ggplot2-Tutorial-Part2-Customizing-Theme-With-R-Code.html

The above tutorial discusses changing "themes" which affect the appearance of ggplot figs. Lets try changing the x axis text labels so they are slanted . . .

```
gg <- gg + theme(axis.text.x=element_text(size=10,
                                          angle = 70,
                                          vjust=.5))
gg <- gg + ylab("Average Income")

# Try labeling the other axis to something more aethetically pleasing

gg
```


1
2
3
4
5
Technical Degree
Other
Medical
Marketing
Life Sciences
Human Resources
EducationField
0
5000
10000
15000
20000
Average Income
Gender
Female
Male