

Midterm_project

Yuanyuan Lin

2019/10/10

```
#install.packages('psych')
#install.packages('GPArotation')
```

```
library(psych)
library(GPArotation)
library(tidyverse)
```

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

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

```
## -- Conflicts -----
```

```
## x ggplot2::%+%( ) masks psych::%+%( )
## x ggplot2::alpha() masks psych::alpha()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag() masks stats::lag()
```

```
library(esquisse)
library(ggthemes)
library(ggplot2)
```

```
Survey<-read.csv("Survey.csv")
```

```
Survey<-na.omit(Survey)
```

```
#head(Survey)
```

```
Survey1<-Survey[, (1:243)]
```

```
show<-Survey %>%
  filter(!is.na(Survey$V4..Important.in.life..Family)) %>%
  count(Survey$V4..Important.in.life..Family)
```

```
show[1,1]='not answered'
show[2,1]='very important'
show[3,1]='rather important'
show[4,1]='not very important'
show[5,1]='not at all important'
show<-as.data.frame(show)
show
```

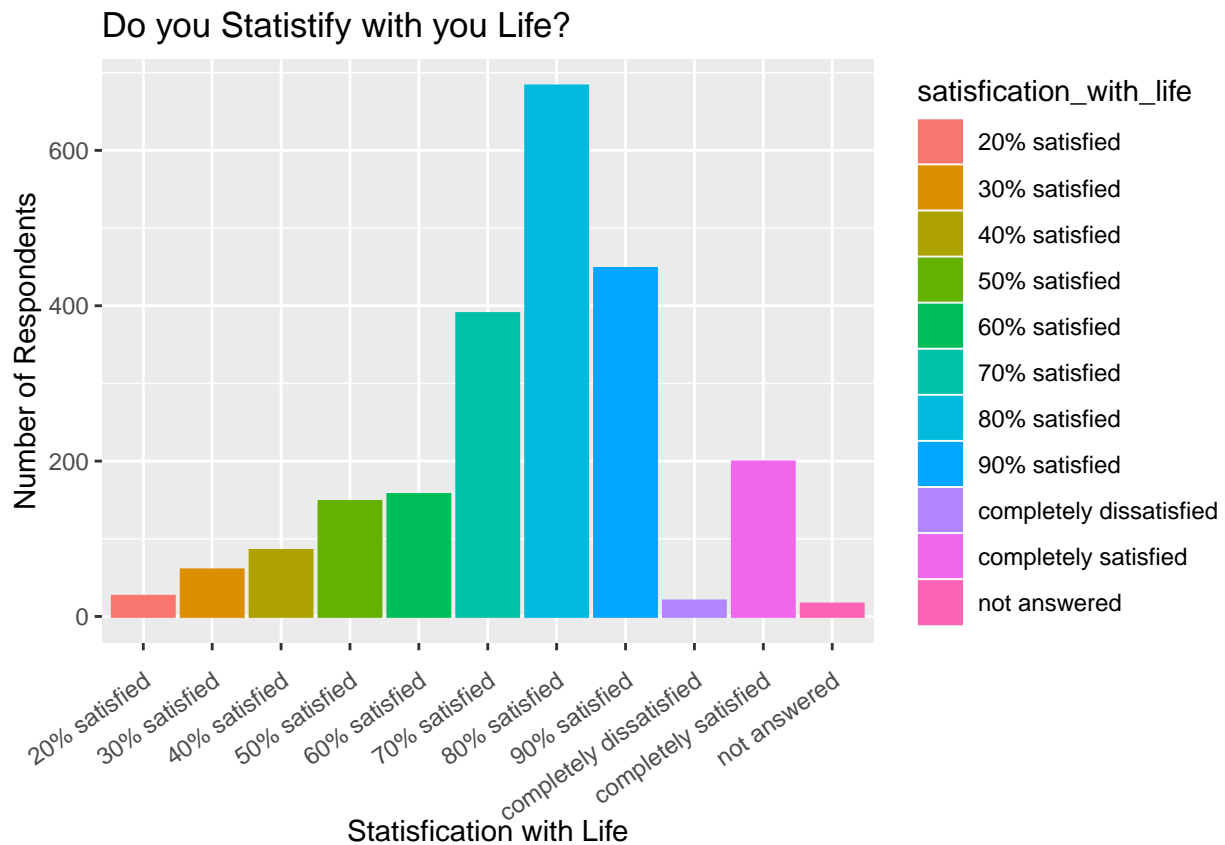
```
## Survey$V4..Important.in.life..Family    n
## 1                not answered    11
## 2                very important 2043
## 3                rather important 147
## 4                not very important 19
## 5                not at all important 12
```

```
show1<-Survey %>%
  filter(!is.na(Survey$V23..Satisfaction.with.your.life)) %>%
  count(Survey$V23..Satisfaction.with.your.life)

show1[1,1]='not answered'
show1[2,1]='completely dissatisfied'
show1[3,1]='20% satisfied'
show1[4,1]='30% satisfied'
show1[5,1]='40% satisfied'
show1[6,1]='50% satisfied'
show1[7,1]='60% satisfied'
show1[8,1]='70% satisfied'
show1[9,1]='80% satisfied'
show1[10,1]='90% satisfied'
show1[11,1]='completely satisfied'
show1<-as.data.frame(show1)
show1<- rename(show1, satisfication_with_life = `Survey$V23..Satisfaction.with.your.life`)
show1
```

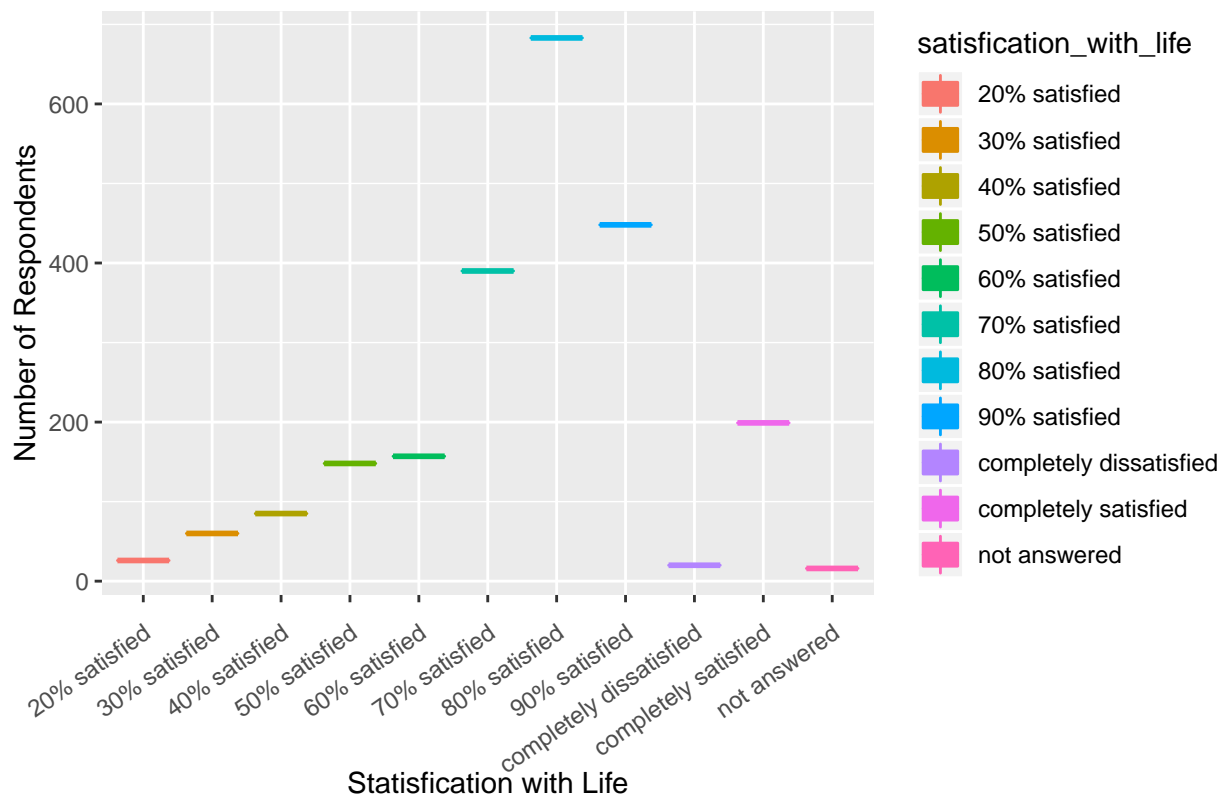
```
## satisfication_with_life    n
## 1                not answered 16
## 2 completely dissatisfied 20
## 3                20% satisfied 26
## 4                30% satisfied 60
## 5                40% satisfied 85
## 6                50% satisfied 148
## 7                60% satisfied 157
## 8                70% satisfied 390
## 9                80% satisfied 683
## 10               90% satisfied 448
## 11 completely satisfied 199
```

```
ggplot(show1) +
  aes(x = satisfication_with_life, fill = satisfication_with_life,
      colour = satisfication_with_life, weight = n) +
  geom_bar() +
  scale_fill_hue() +
  scale_color_hue() +
  theme_gray()+
  theme( axis.text.x = element_text(angle=35, hjust=1, vjust=0.9))+
  labs(x = "Statisfication with Life", y = "Number of Respondents") +
  ggtitle("Do you Statistify with you Life?")
```



```
ggplot(show1) +
  aes(x = satisfication_with_life, y = n, fill = satisfication_with_life, colour = satisfication_with_li) +
  geom_boxplot() +
  scale_fill_hue() +
  scale_color_hue() +
  theme_gray() +
  theme(axis.text.x = element_text(angle=35, hjust=1, vjust=0.9)) +
  labs(x = "Statisfication with Life", y = "Number of Respondents") +
  ggtitle("Box?")
```

Box?



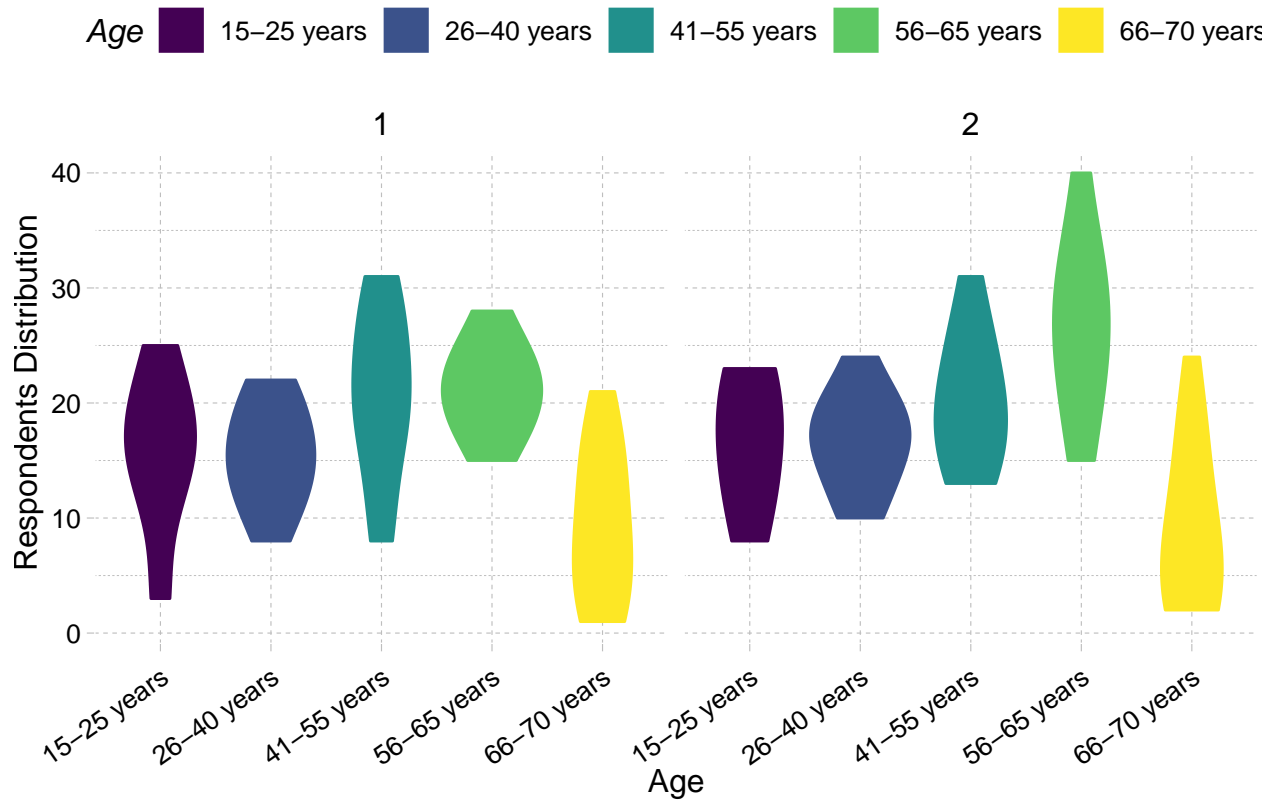
```
show2<-Survey %>%
  filter(!is.na(Survey$V240..Sex) & !is.na(Survey$V242..Age)) %>%
  group_by(Survey$V240..Sex, Survey$V242..Age) %>%
  count()
show2$`Survey$V242..Age` <- cut(show2$`Survey$V242..Age`,
  breaks = c(-Inf, 25, 40, 55, 65, Inf),
  labels = c("15-25 years", "26-40 years", "41-55 years", "56-65 years", "66-70 years"),
  right = FALSE)
show2<-as.data.frame(show2)

#show2[show2 == '1']<-"Male"
#show2[show2 == '2']<-"Female"
#show2
```

```
show2<- rename(show2, Age = `Survey$V242..Age`)
#show2
ggplot(show2) +
  aes(x = Age, y = n, fill = Age, colour = Age) +
  geom_violin(adjust = 1.6, scale = "area") +
  scale_fill_viridis_d(option = "viridis") +
  scale_color_viridis_d(option = "viridis") +
  theme_pander() +
  theme(legend.position = "top") +
  facet_wrap(vars(`Survey$V240..Sex`))+
  theme( axis.text.x = element_text(angle=35, hjust=1, vjust=0.9))+
  labs(x = "Age", y = " Respondents Distribution") +
```

```
ggtitle("Distribution of Number of Respondents among Age Group in Male/Female")
```

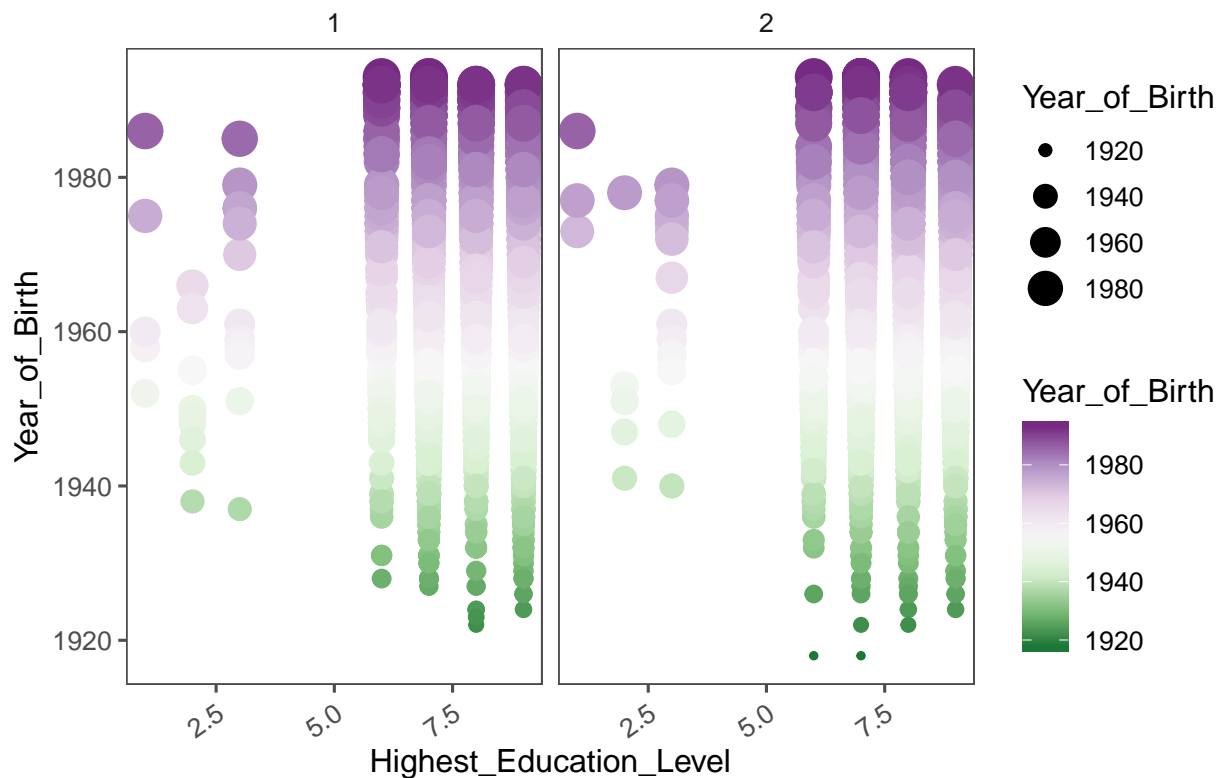
Distribution of Number of Respondents among Age Group in Male



```
Survey<- rename(Survey, Highest_Education_Level = V248..Highest.educational.level.attained)
Survey<- rename(Survey, Year_of_Birth = V241..Year.of.birth)
```

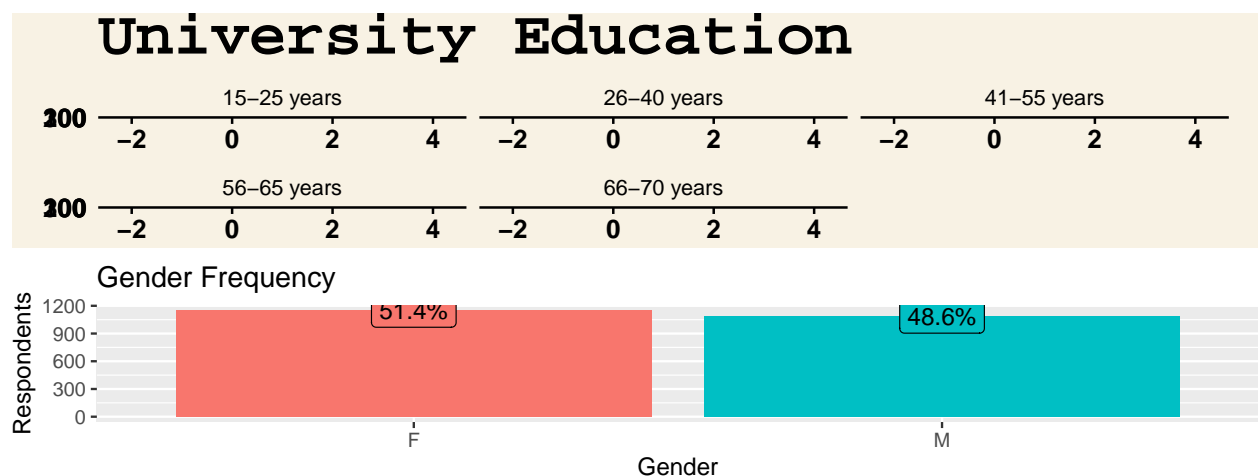
```
knitr::opts_chunk$set(echo = FALSE, fig.height=1.5, fig.width=8)
ggplot(Survey) +
  aes(x = Highest_Education_Level, y = Year_of_Birth, colour = Year_of_Birth, size = Year_of_Birth) +
  geom_point() +
  scale_color_distiller(palette = "PRGn") +
  theme_few() +
  facet_wrap(vars(V240..Sex))+
  theme( axis.text.x = element_text(angle=35, hjust=1, vjust=0.9))+
  ggtitle("Scatter Plot between Highest Education Level and Year of Birth in Female/Male")
```

Scatter Plot between Highest Education Level and Year of Birth in Female



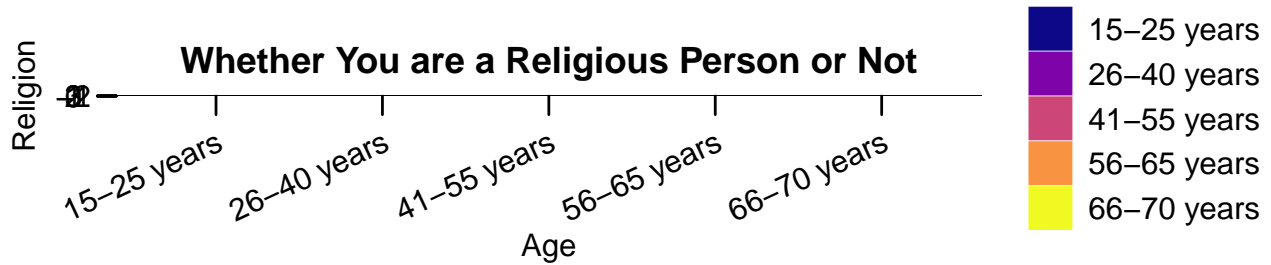
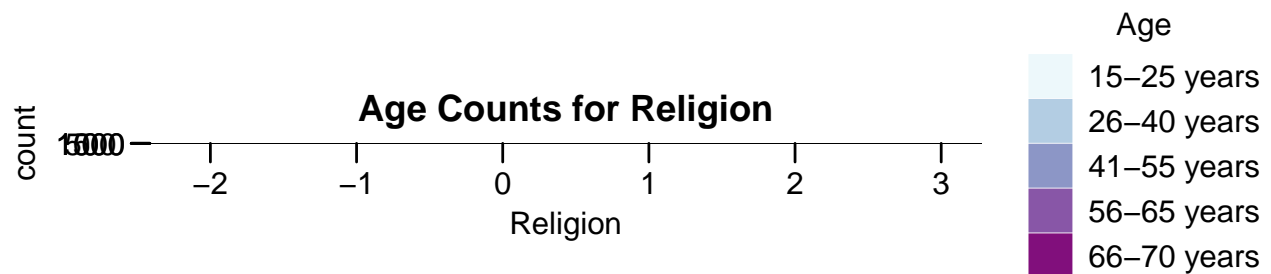
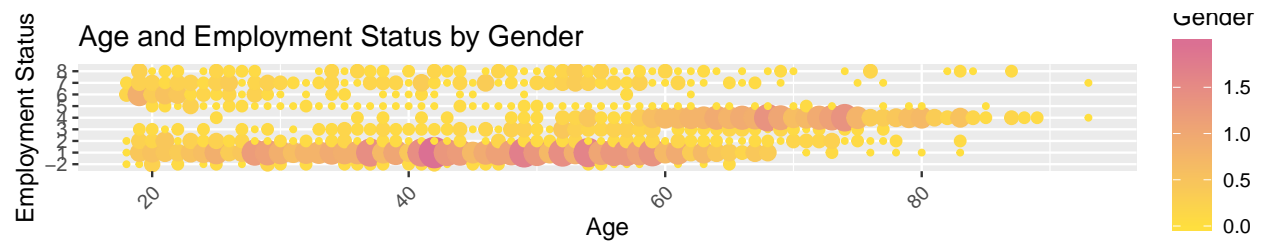
```
## # A tibble: 2 x 2
##   Hard_Work    n
##   <int> <int>
## 1       1 1463
## 2       2  769
```

```
##   Independence    n Hard_Work    n.1
## 1             1 1178           1 1463
## 2             2 1054           2  769
```



```
## Warning: `as_data_frame()` is deprecated, use `as_tibble()` (but mind the new semantics).
```

This warning is displayed once per session.



```
## function (x = character(), levels, labels = levels, exclude = NA,
##   ordered = is.ordered(x), nmax = NA)
## {
##   if (is.null(x))
##     x <- character()
##   nx <- names(x)
##   if (missing(levels)) {
##     y <- unique(x, nmax = nmax)
##     ind <- order(y)
##     levels <- unique(as.character(y)[ind])
##   }
##   force(ordered)
##   if (!is.character(x))
##     x <- as.character(x)
##   levels <- levels[is.na(match(levels, exclude))]
##   f <- match(x, levels)
##   if (!is.null(nx))
##     names(f) <- nx
##   if (missing(labels)) {
##     levels(f) <- as.character(levels)
##   }
##   else {
##     nlab <- length(labels)
##     if (nlab == length(levels)) {
##       nlevs <- unique(xlevs <- as.character(labels))
##       at <- attributes(f)
```

```

##           at$levels <- nlevs
##           f <- match(xlevs, nlevs)[f]
##           attributes(f) <- at
##       }
##       else if (nlab == 1L)
##           levels(f) <- paste0(labels, seq_along(levels))
##       else stop(gettextf("invalid 'labels'; length %d should be 1 or %d",
##           nlab, length(levels)), domain = NA)
##   }
##   class(f) <- c(if (ordered) "ordered", "factor")
##   f
## }
## <bytecode: 0x7f87680f2d90>
## <environment: namespace:base>

```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.