

Actividad 6

Natalia da Silva

1/4/2018

Ejercicio 1

1. Crear una nuevo repositorio en GitHub llamado Actividad 6.
2. Crear un proyecto de RStudio y conectarlo con su nuevo repositorio en GitHub
3. Abrir un archivo de Rmarkdown que compile a pdf con la solución de la Actividad 6, puede usar como base el .Rmd que está disponible en EVA.
4. Realizar el primer `commit` y `push` con la Actividad 6 en su nuevo repositorio de GitHub (sólo el .Rmd).

Ejercicio 2

Cargar la librería `tidyverse` y con los datos `mpg` usar las funciones que vimos de `dplyr` para transformar los datos de la siguiente forma.

1. Seleccionar el conjunto de autos del 2008 que tienen `hwy` mayor a 31. ¿Cuántos autos son?
2. Seleccionar el conjutno de autos que tienen `cty` igual a 9 ó `hwy` es mayor a 90. ¿Cuántos autos son?
3. Selecciona la variable `displ` y `hwy`
4. Seleccioná desde la variable `trans` hasta la variable `hwy`
5. Crear una nueva variable que transforme `hwy` (millas por galón) en litros/100 km Primero tenés que saber la conversión de galones a litros y de millas a kilómetros

1 `gallón = 3.78541 litros`

1 `milla = 1.60934 km`

Ejercicio 2

Parte 1

```
require(ggplot2)
require(dplyr)
filter(mpg, year==2008, hwy>31)
```

```
## # A tibble: 6 x 11
##   manufacturer model displ  year   cyl trans drv   cty   hwy fl   class
##   <chr>          <chr> <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
## 1 honda        civic  1.8  2008     4 manu~ f     26    34 r   subc~
## 2 honda        civic  1.8  2008     4 auto~ f     25    36 r   subc~
## 3 honda        civic  1.8  2008     4 auto~ f     24    36 c   subc~
## 4 nissan        alti~  2.5  2008     4 manu~ f     23    32 r   mids~
## 5 toyota        coro~  1.8  2008     4 manu~ f     28    37 r   comp~
## 6 toyota        coro~  1.8  2008     4 auto~ f     26    35 r   comp~
```

Son 6 autos los que cumplen esas condiciones

Bien, si fueran más podrías usar ‘summarise’ para contar la cantidad

Parte 2

```
filter(mpg, cty==9 | hwy>90)
```

```
## # A tibble: 5 x 11
##   manufacturer model displ  year   cyl trans drv   cty   hwy fl   class
##   <chr>          <chr> <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
## 1 dodge        dako~  4.7  2008     8 auto~ 4     9    12 e   pick~
## 2 dodge        dura~  4.7  2008     8 auto~ 4     9    12 e   suv
## 3 dodge        ram ~  4.7  2008     8 auto~ 4     9    12 e   pick~
## 4 dodge        ram ~  4.7  2008     8 manu~ 4     9    12 e   pick~
## 5 jeep         gran~  4.7  2008     8 auto~ 4     9    12 e   suv
```

Son 5 autos los que cumplen esas condiciones

Bien

Parte 3

```
select(mpg, displ, hwy)
```

```
## # A tibble: 234 x 2
##   displ  hwy
##   <dbl> <int>
## 1  1.8    29
## 2  1.8    29
## 3  2      31
## 4  2      30
## 5  2.8    26
## 6  2.8    26
## 7  3.1    27
## 8  1.8    26
## 9  1.8    25
## 10 2      28
## # ... with 224 more rows
```

Bien

Parte 4

```
select(mpg, trans:hwy)
```

```
## # A tibble: 234 x 4
##   trans      drv   cty   hwy
##   <chr>    <chr> <int> <int>
## 1 auto(l5)  f      18    29
## 2 manual(m5) f      21    29
## 3 manual(m6) f      20    31
## 4 auto(av)  f      21    30
## 5 auto(l5)  f      16    26
## 6 manual(m5) f      18    26
## 7 auto(av)  f      18    27
## 8 manual(m5) 4      18    26
## 9 auto(l5)  4      16    25
## 10 manual(m6) 4      20    28
## # ... with 224 more rows
```

Bien!

Parte 5

```
mpg= mutate(mpg, hwy_n = ((1/hwy) * (3.785)*(62.150)))  
select(mpg, hwy, hwy_n)
```

```
## # A tibble: 234 x 2  
##       hwy hwy_n  
##   <int> <dbl>  
## 1     29  8.11  
## 2     29  8.11  
## 3     31  7.59  
## 4     30  7.84  
## 5     26  9.05  
## 6     26  9.05  
## 7     27  8.71  
## 8     26  9.05  
## 9     25  9.41  
## 10    28  8.40  
## # ... with 224 more rows
```

Bien!

Ejercicio 3

Parte 1

```
require(productplots)

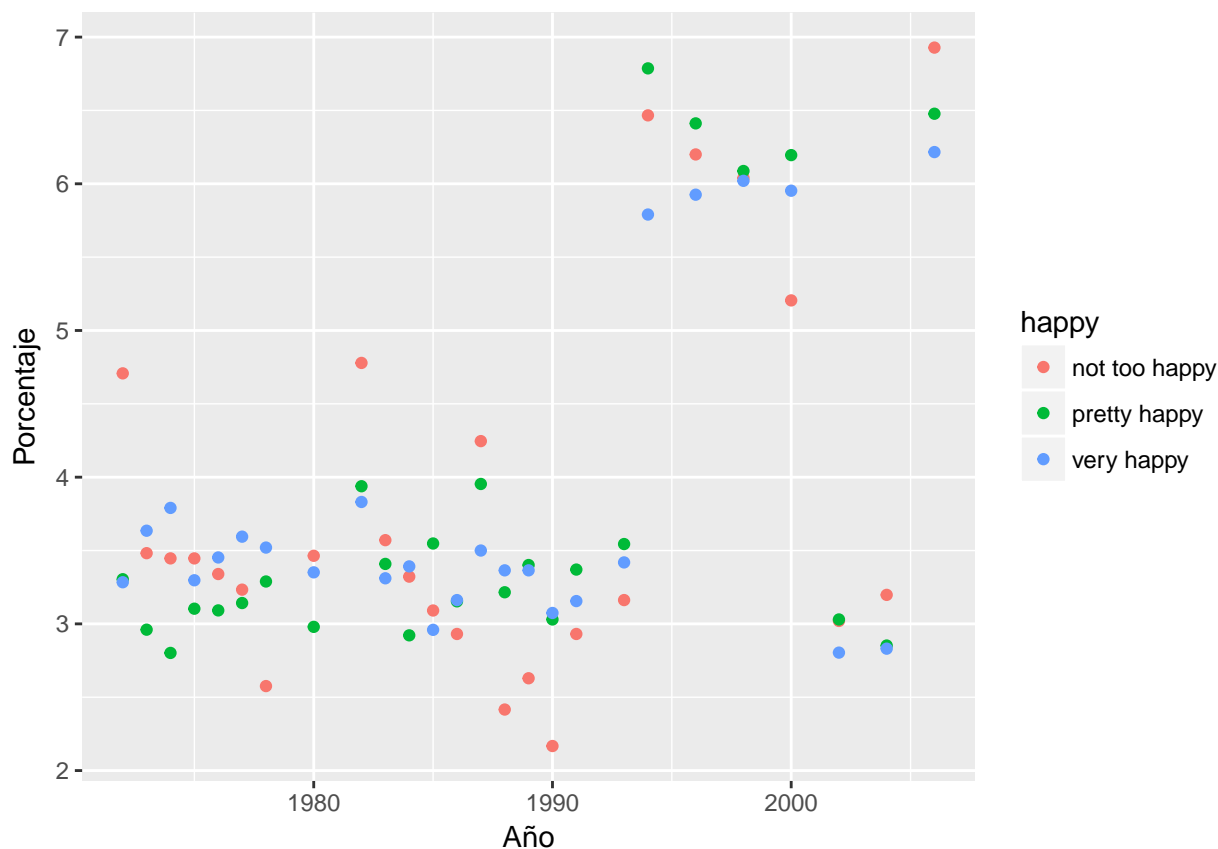
## Loading required package: productplots
happy %>%
  group_by(sex, age, happy)%>%
  summarise(happy_n=n())%>%
  mutate(happy_percent=happy_n/sum(happy_n))

## # A tibble: 584 x 5
## # Groups:   sex, age [146]
##   sex    age happy    happy_n happy_percent
##   <fct> <dbl> <fct>      <int>      <dbl>
## 1 male    18 not too happy    11        0.122
## 2 male    18 pretty happy    47        0.522
## 3 male    18 very happy    16        0.178
## 4 male    18 <NA>         16        0.178
## 5 male    19 not too happy    45        0.129
## 6 male    19 pretty happy   217        0.62
## 7 male    19 very happy    70         0.2
## 8 male    19 <NA>         18       0.0514
## 9 male    20 not too happy    43        0.126
## 10 male   20 pretty happy   208        0.610
## # ... with 574 more rows
```

Cuando puse a lo largo de todos los años esperaba que en el agrupamiento usaran year en vez de edad. A su vez hay que tener cuidado con 'group_by' ya que hay tener cuidado donde suma 100%. Ver solución.

Parte 2

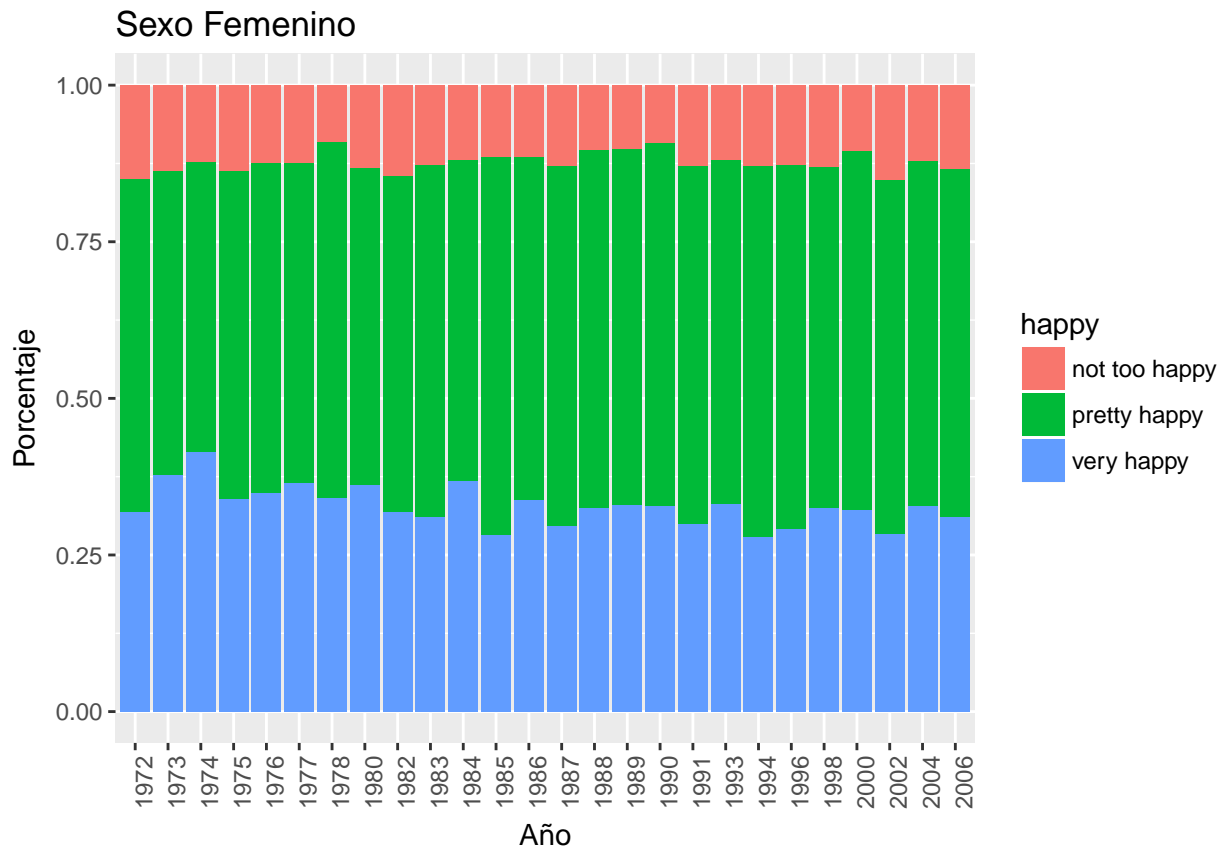
```
happy%>%  
  filter(!is.na(happy)) %>%  
  group_by(happy, year) %>%  
  summarise(happy_n= n()) %>%  
  mutate(happy_percent = happy_n / sum(happy_n) * 100) %>%  
  ggplot(aes(x = year, y = happy_percent, colour = happy)) +  
  geom_point()+  
  labs(x="Año", y="Porcentaje")
```



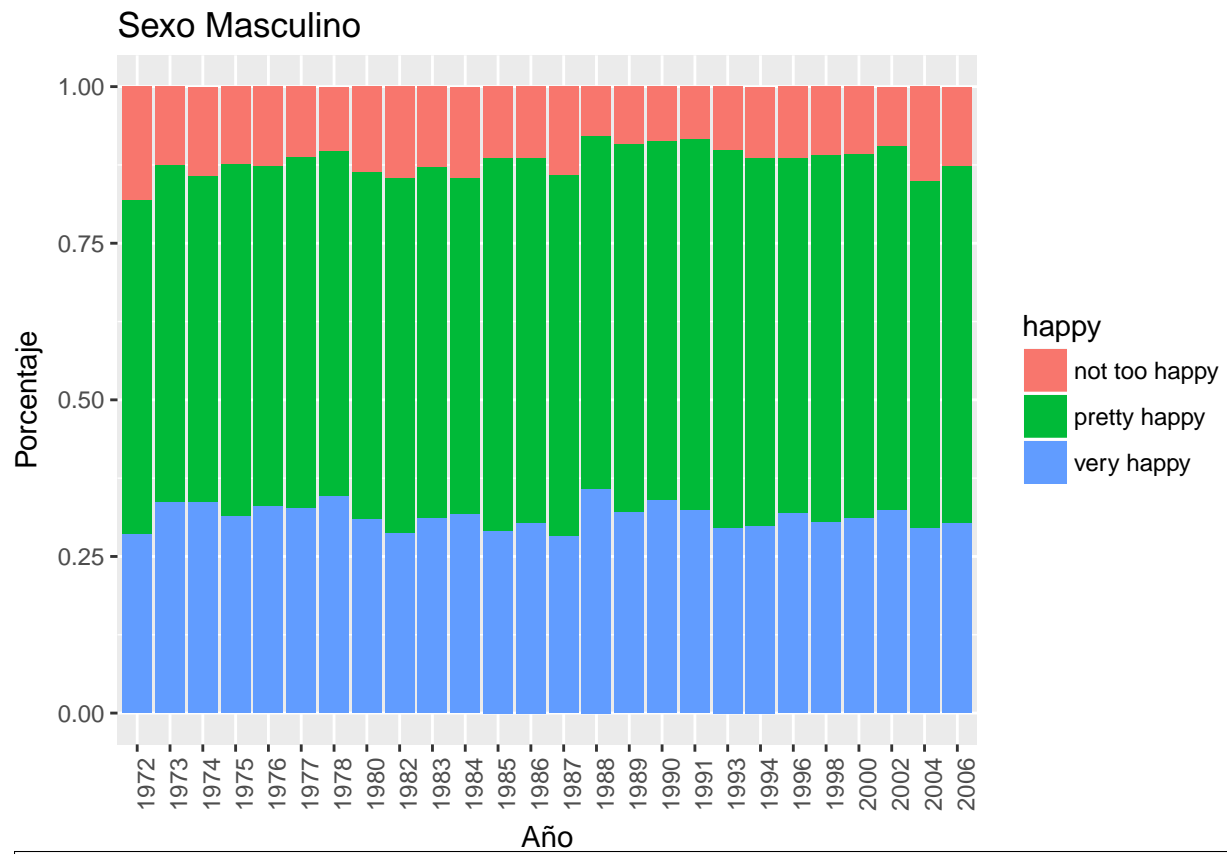
Esperaba el mismo agrupamiento que el anterior tendría que haber 6 puntos en cada año. Qué suma 100% en tu visualización? hay que tener cuidado con el agrupamiento , si corrés hasta summarise vas a ver que la última variable con la que agrupaste no la toma en cuenta y define Groups solo con happy. Aunque no lo pedí en agregaría un comentario interesante de la figura, el comentario te ayudaría a ver si tiene o no sentido lo que ves y lo que estás dibujando.

Parte 3

```
happy%>%
  filter(!is.na(happy), sex=="female")%>%
  ggplot()+
  geom_bar(aes(x = as.character(year), fill = happy), position = "fill")+
  ggtitle("Sexo Femenino")+
  theme(axis.text.x = element_text(angle=90))+
  labs(x="Año", y="Porcentaje")
```



```
happy%>%
  filter(!is.na(happy), sex=="male")%>%
  ggplot()+
  geom_bar(aes(x = as.character(year), fill = happy), position = "fill")+
  ggtitle("Sexo Masculino")+
  theme(axis.text.x = element_text(angle=90))+
  labs(x="Año", y="Porcentaje")
```

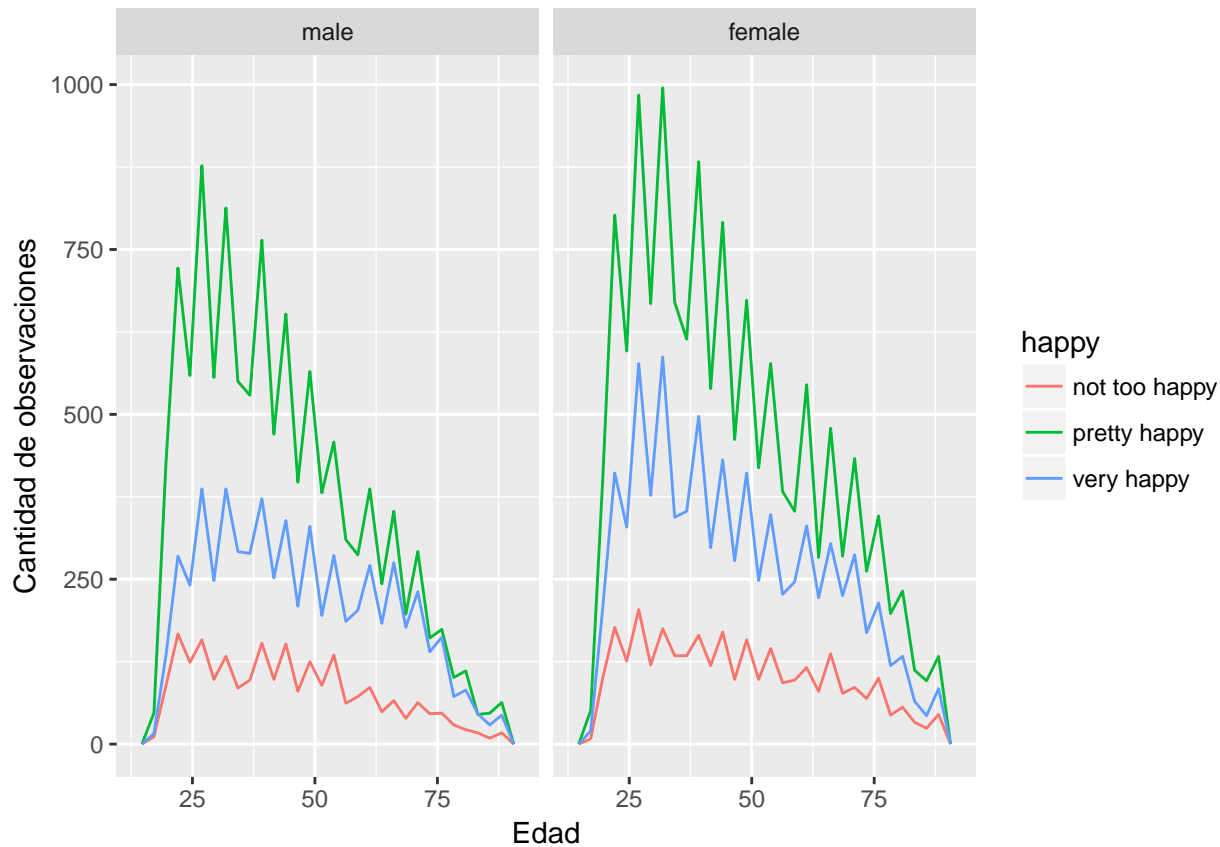


En este caso usando 'facet_wrap' facilitaría la comparación porque veríamos los dos en el mismo display gráfico. Se puede hacer sin incluir año.

Parte 4

```
happy%>%  
  filter(!is.na(happy), !is.na(age))%>%  
  ggplot(aes(x = age, colour=happy))+  
  geom_freqpoly()+  
  facet_wrap(~sex)+  
  labs(x="Edad", y= "Cantidad de observaciones")
```

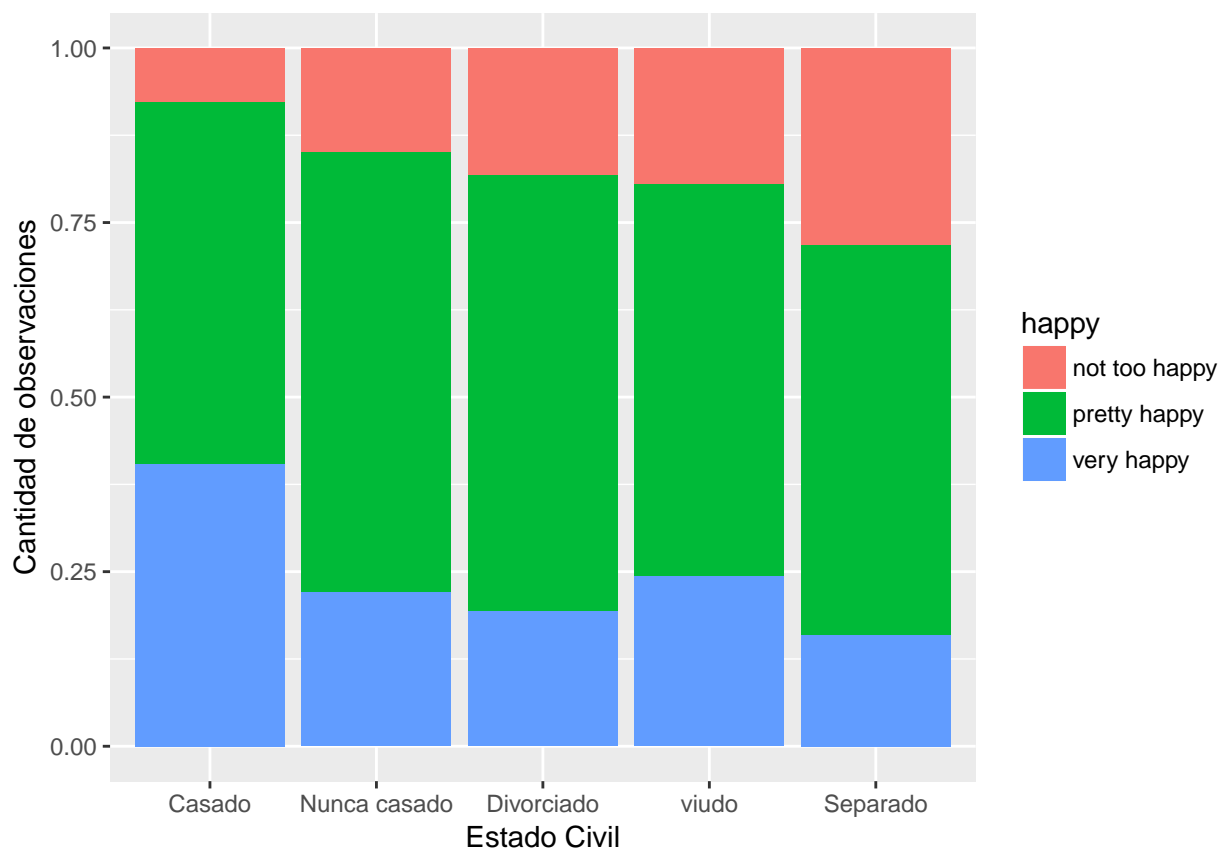
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.



Parte 5

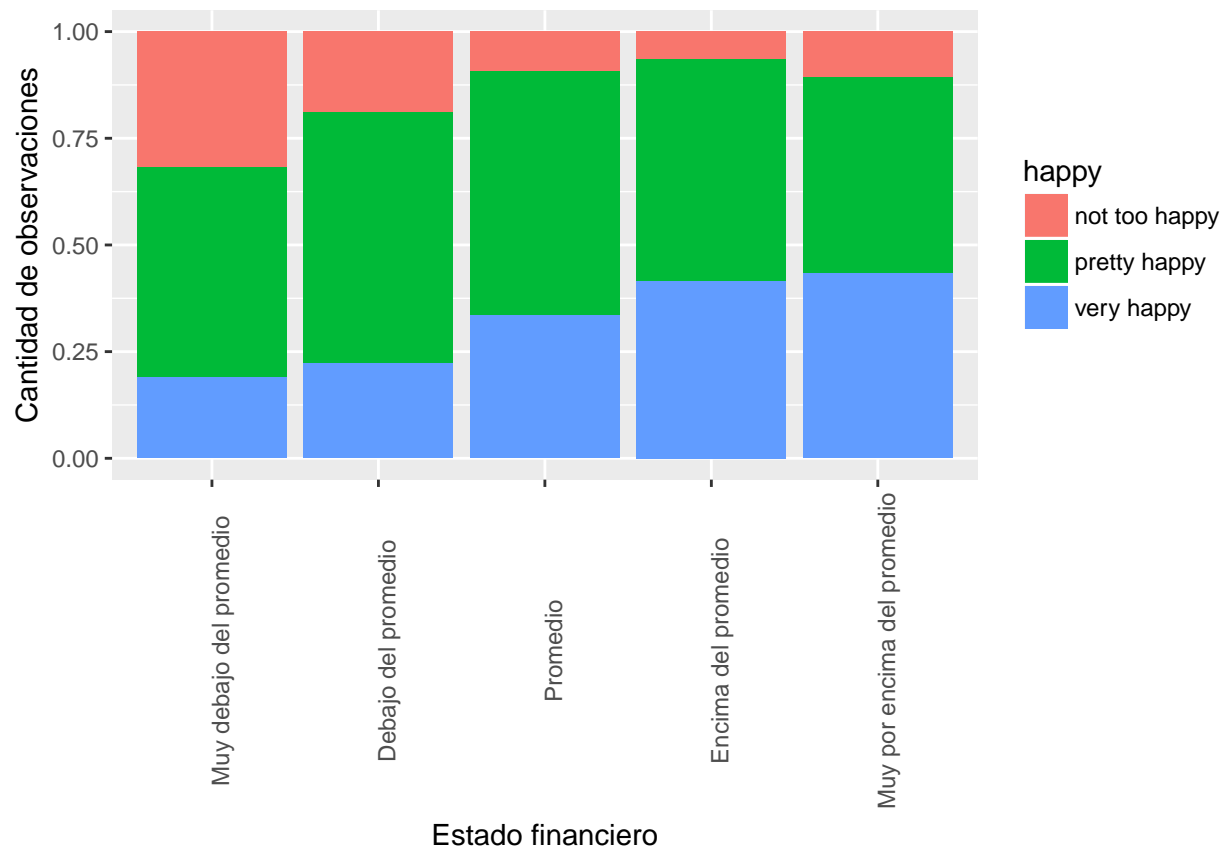
```
levels(happy$marital)= c("Casado", "Nunca casado", "Divorciado", "viudo", "Separado")
```

```
happy%>%  
  filter(!is.na(happy), !is.na(marital))%>%  
  ggplot(aes(x = marital, fill=happy))+  
  geom_bar(position = "fill")+  
  labs(x="Estado Civil", y="Cantidad de observaciones")
```



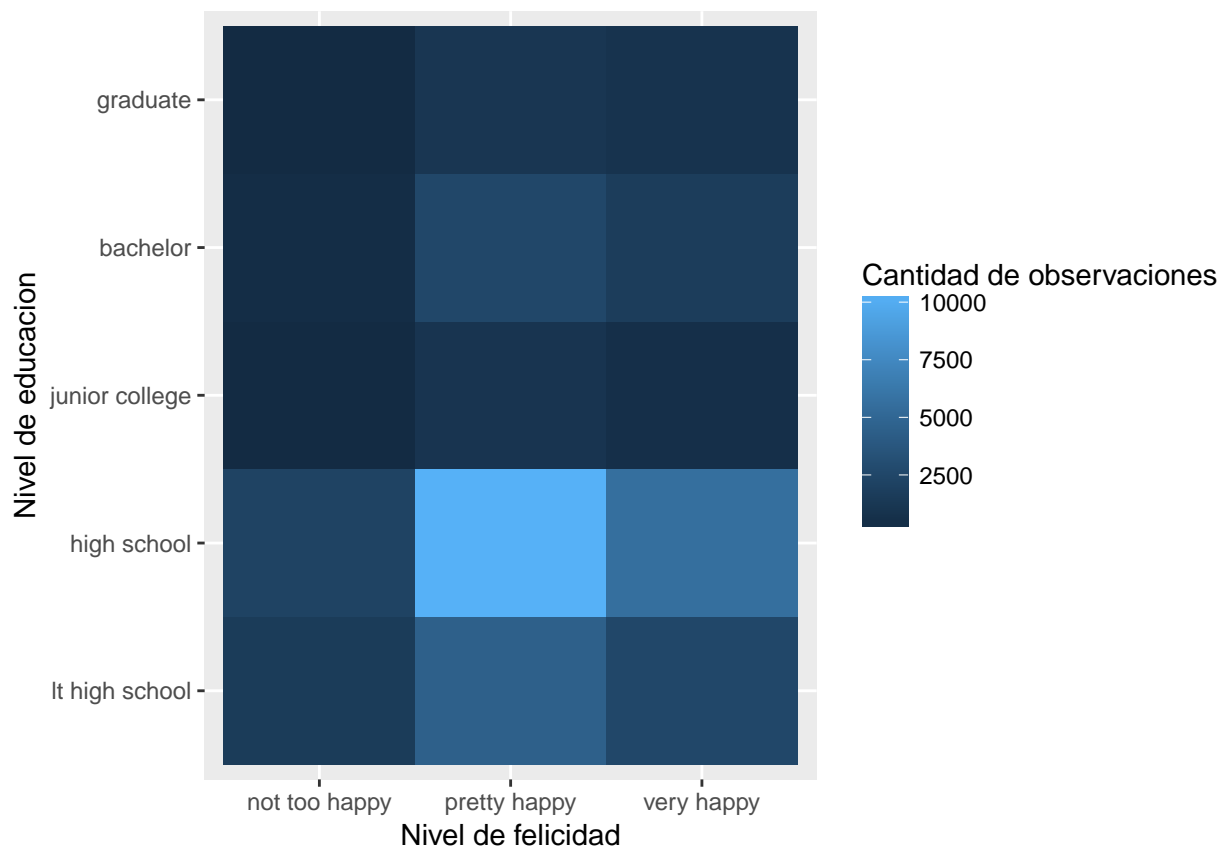
```
levels(happy$finrela)=c("Muy debajo del promedio", "Debajo del promedio", "Promedio", "Encima del promedio")
```

```
happy%>%  
  filter(!is.na(happy), !is.na(finrela))%>%  
  ggplot(aes(x = finrela, fill=happy))+  
  geom_bar(position = "fill")+  
  theme(axis.text.x = element_text(angle=90))+  
  labs(x="Estado financiero", y="Cantidad de observaciones")
```



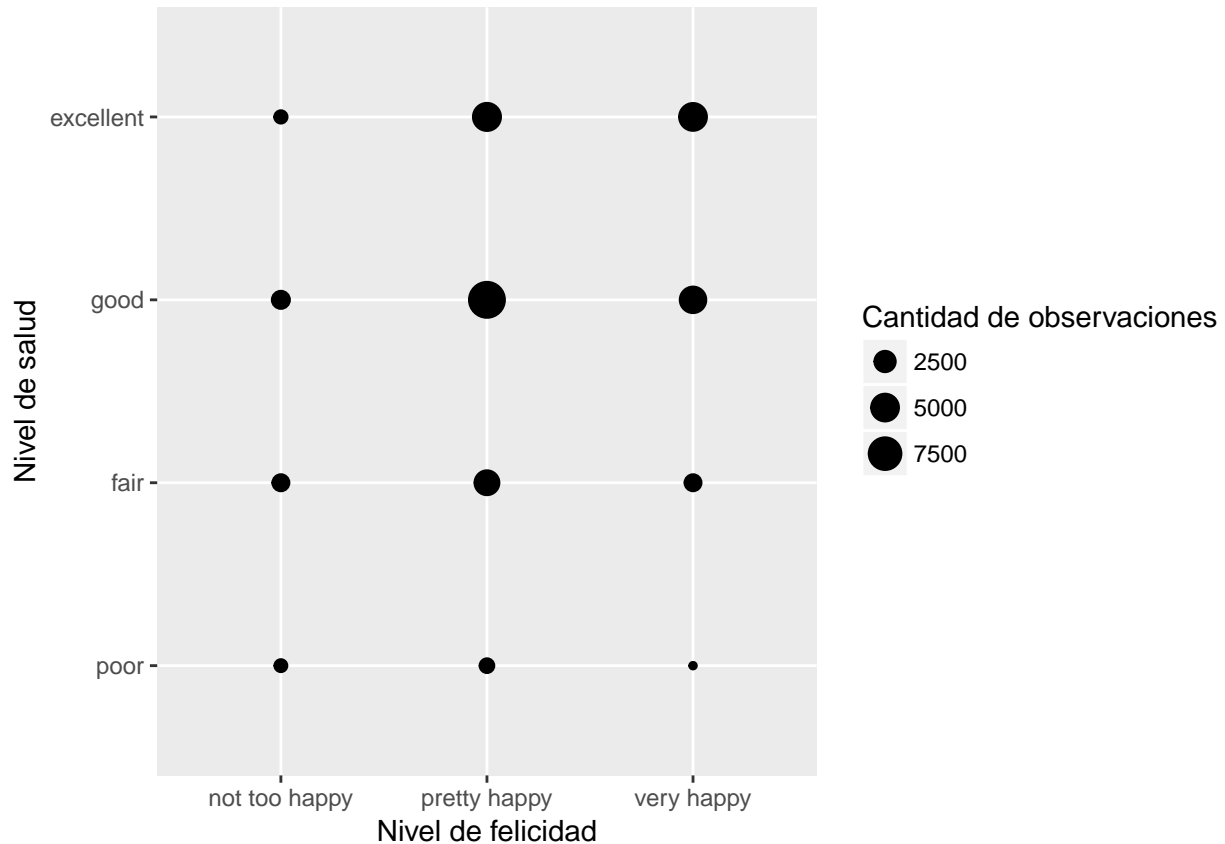
Parte 6

```
happy %>% na.omit() %>%  
  group_by( happy, degree) %>%  
  summarise(n = n()) %>%  
  ggplot(aes(x = happy, y = degree, fill=n)) +  
  geom_tile()+  
  labs(x="Nivel de felicidad", y="Nivel de educacion", fill="Cantidad de observaciones")
```



Parte 7

```
happy %>% na.omit() %>%  
  group_by( happy, health) %>%  
  summarise(n = n()) %>%  
  ggplot(mapping = aes(x = happy, y = health, size=n)) +  
  geom_count()+  
  labs(x="Nivel de felicidad", y="Nivel de salud", size="Cantidad de observaciones")
```



En el ejercicio 3 no alcanza con proponer una visualización si no interpretás y respondés las preguntas planteadas. Tu código y trabajo en general está ordenado y prolijo pero le faltan comentarios. Ver solución. Puntos 8/10