Práctica 2: Limpieza y análisis de datos

Pedro Uceda Martínez, Pablo Campillo Sánchez

30 de diciembre, 2020

1. Descripción del dataset

Durante esta práctica vamos a tratar el *dataset* base de la competición **Titanic - Machine Learning from Disaster**. En este conjunto de datos se nos presenta, para cada pasajero del tan famoso trasatlántico, sus datos personales más importantes, así como otros relacionados con su embarque en el Titanic, y si finalmente sobrevivieron al naufragio del mismo.

De este modo, este estudio es interesante dado que examinaremos qué posibles factores pudieron influir en la supervivencia de los pasajeros. Así, podremos, por ejemplo, ver si solamente la clase del billete, el género (mujeres) y la edad (niños) condicionaron que un viajero se salvase tal y como hemos visto en la gran pantalla o bien hubiera habido otros factores que pudieran haber determinado la supervivencia del pasajero, como el número de billete.

Las variables de las que disponemos, para cada pasajero, son:

- **PassengerId**: Identificador artifical del pasajero.
- **Survived**: Si sobrevivió (1) o no (0).
- **Pclass**: Clase del pasaje.
- **Name**: Nombre del pasajero.
- **sex**: Sexo del viajero.
- **Age**: Edad, en años.
- **SibSp**: Número de hermanos o esposas a bordo del Titanic
- **Parch**: Número de padres / hijos a bordo del Titanic
- **ticket**: Número de ticket
- **fare**: Tarifa del pasaje
- **cabin**: Número de camarote
- **embarked**: Puerto desde el que embarcó el pasajero. Las posibles opciones son: Cherbourg(C), Queenstown(Q) o Southampton(s).

2. Integración y selección de los datos de interés a analizar.

Los datos a procesar provienen de una única fuente, por ello, no es necesario realizar la fase de integración o fusionado de los datos. En este apartado, primero se cargarán los datos y se hará una exploración inicial de los mismos para tener una idea más clara de los mismos y, posteriormente, se procede a seleccionar los datos de interés y a generar nuevas características que puedan resultar interesantes para el análisis posterior.

2.1 Exploración de los datos (screening)

A continuación procedemos a cargar el **dataset**, sin **factors**, para evitar tratar los nombres de los pasajeros como tales.

```r
ds <- read.csv(file = "train.csv", header=TRUE, stringsAsFactors=FALSE)
str(ds)
```

```
## 'data.frame':    891 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
##  $ Sex        : chr  "male" "female" "female" "female" ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  "" "C85" "" "C123" ...
##  $ Embarked   : chr  "S" "C" "S" "S" ...
```

Como se puede observar, el **dataset** contiene 891 registros y 12 atributos. Tenemos las variables cuantitativas PassengerId, Survived, Pclass, Age, SibSp, Parch y Fare, todas tratadas como int o num. También están las variables cualitativas Ticket, PClass, Sex y Cabin, cargadas como cadena de caracteres.

Para más claridad de los datos, procedemos a realizar las siguientes transformaciones: - Transformamos el campo dicotómico Survived a Yes(1) y Not(0). - Transformamos el campo cualitativo categórico Embarked a un factor con 3 posibles valores, cada uno con el nombre del puerto. - Transformamos el campo dicotómico Sex en vez de cadena.

```r
ds$Survived <- factor(ds$Survived, levels=sort(c(0,1)), labels = c("Not", "Yes"))
ds$Embarked <- factor(ds$Embarked, levels=sort(c("C", "Q", "S")), labels = c("Cherbourg", "Queenstown",
ds$Sex <- factor(ds$Sex)
str(ds)
```

```
## 'data.frame':    891 obs. of  12 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Survived   : Factor w/ 2 levels "Not","Yes": 1 2 2 2 1 1 1 1 2 2 ...
##  $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
##  $ Sex        : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  "" "C85" "" "C123" ...
##  $ Embarked   : Factor w/ 3 levels "Cherbourg","Queenstown",..: 3 1 3 3 3 2 3 3 3 1 ...
```

Para hacernos una idea de las características, vamos a mostrar las estadístcas básicas:

```r
summary(ds)
```

```
##   PassengerId    Survived      Pclass          Name                Sex
##  Min.   :  1.0   Not:549   Min.   :1.000   Length:891         female:314
##  1st Qu.:223.5   Yes:342   1st Qu.:2.000   Class :character   male  :577
##  Median :446.0             Median :3.000   Mode  :character
##  Mean   :446.0             Mean   :2.309
##  3rd Qu.:668.5             3rd Qu.:3.000
##  Max.   :891.0             Max.   :3.000
##
##       Age            SibSp           Parch            Ticket
##  Min.   : 0.42   Min.   :0.000   Min.   :0.0000   Length:891
##  1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000   Class :character
##  Median :28.00   Median :0.000   Median :0.0000   Mode  :character
##  Mean   :29.70   Mean   :0.523   Mean   :0.3816
##  3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
##  Max.   :80.00   Max.   :8.000   Max.   :6.0000
##  NA's   :177
##       Fare           Cabin                 Embarked
##  Min.   :  0.00   Length:891         Cherbourg  :168
##  1st Qu.:  7.91   Class :character   Queenstown : 77
##  Median : 14.45   Mode  :character   Southampton:644
##  Mean   : 32.20                      NA's       :  2
##  3rd Qu.: 31.00
##  Max.   :512.33
##
```

La información más relevante es:

- **Survived**: Hay más gente que falleció que sobrevivió.
- **Pclass**: Lo más común es tercera clases (Median).
- **Sex**: En el barco viajaban el doble de hombres que de mujeres.
- **age**: especifica la edad en años. Podemos ver que el mínimo es 0.42 años, así que se contemplan bebés. La persona más anciana tenía 80 años y la media de edad estaba en torno a los 30 años.
- **SibSp**: Lo más común es ir sin hermanos ni mujer.
- **Parch**: Es menos común todavía ir con descendientes o ascendientes.
- **Fare**: La media del precio del billete es 32.2 y la mediana 14. Esto indica que hay mucha disparidad de precios, siendo el máximo 512.
- **Embarked**: La mayoría embarcaron de Southampton, luego de Cherbourg y unos pocos de Queenstown.

Por último, hacemos una inspección visual de los campos que menos sabemos sobre ellos: Ticket y Cabin.

La codificación del billete (Ticket) parece que sigue diferentes patrones y además, hay viajeros que comparten el ticket ya que si los ordenamos, podemos comprobar que estos se repiten:

```r
sort(ds$Ticket)[1:10]
```

```
##  [1] "110152" "110152" "110152" "110413" "110413" "110413" "110465" "110465"
##  [9] "110564" "110813"
```

Si comprobamos los campos únicos, vemos que pasa de 891 a 681 valores diferentes.

```r
length(distinct(ds, Ticket)$Ticket)
```

```
## [1] 681
```

Además, el que un ticket se repita no depende de su tipo:

```r
aux <- count(ds, Ticket)
aux[order(aux[,2], decreasing = TRUE), ][1:10, ]
```

```
##           Ticket n
## 81          1601 7
## 334       347082 7
## 569     CA. 2343 7
## 250      3101295 6
## 338       347088 6
## 567      CA 2144 6
## 481       382652 5
## 622 S.O.C. 14879 5
## 34        113760 4
## 38        113781 4
```

Suponemos que se puede comprar un mismo billete para varias personas. ¿Compartirán el camarote? ¿Serán familia? Veamos los datos de estos 10.

Ticket 1601:

```
select(ds[ds$Ticket == "1601", ], Name, Pclass, Fare, Cabin, Embarked, Sex, Age, SibSp, Parch)
```

```
##                Name Pclass    Fare Cabin    Embarked  Sex Age SibSp Parch
## 75    Bing, Mr. Lee      3 56.4958       Southampton male  32     0     0
## 170   Ling, Mr. Lee      3 56.4958       Southampton male  28     0     0
## 510  Lang, Mr. Fang      3 56.4958       Southampton male  26     0     0
## 644 Foo, Mr. Choong      3 56.4958       Southampton male  NA     0     0
## 693    Lam, Mr. Ali      3 56.4958       Southampton male  NA     0     0
## 827    Lam, Mr. Len      3 56.4958       Southampton male  NA     0     0
## 839 Chip, Mr. Chang      3 56.4958       Southampton male  32     0     0
```

Ticket 347082:

```
select(ds[ds$Ticket == "347082", ], Name, Pclass, Fare, Cabin, Embarked, Sex, Age, SibSp, Parch)
```

```
##                                                          Name Pclass   Fare
## 14                                Andersson, Mr. Anders Johan      3 31.275
## 120                         Andersson, Miss. Ellis Anna Maria      3 31.275
## 542                      Andersson, Miss. Ingeborg Constanzia      3 31.275
## 543                         Andersson, Miss. Sigrid Elisabeth      3 31.275
## 611 Andersson, Mrs. Anders Johan (Alfrida Konstantia Brogren)      3 31.275
## 814                        Andersson, Miss. Ebba Iris Alfrida      3 31.275
## 851                   Andersson, Master. Sigvard Harald Elias      3 31.275
##     Cabin    Embarked    Sex Age SibSp Parch
## 14        Southampton   male  39     1     5
## 120       Southampton female   2     4     2
## 542       Southampton female   9     4     2
## 543       Southampton female  11     4     2
## 611       Southampton female  39     1     5
## 814       Southampton female   6     4     2
## 851       Southampton   male   4     4     2
```

Ticket CA. 2343:

```
select(ds[ds$Ticket == "CA. 2343", ], Name, Pclass, Fare, Cabin, Embarked, Sex, Age, SibSp, Parch)
```

```
##                                  Name Pclass  Fare Cabin    Embarked    Sex Age
## 160        Sage, Master. Thomas Henry      3 69.55       Southampton   male  NA
## 181      Sage, Miss. Constance Gladys      3 69.55       Southampton female  NA
## 202               Sage, Mr. Frederick      3 69.55       Southampton   male  NA
## 325          Sage, Mr. George John Jr      3 69.55       Southampton   male  NA
## 793           Sage, Miss. Stella Anna      3 69.55       Southampton female  NA
## 847          Sage, Mr. Douglas Bullen      3 69.55       Southampton   male  NA
## 864 Sage, Miss. Dorothy Edith "Dolly"      3 69.55       Southampton female  NA
##     SibSp Parch
## 160     8     2
## 181     8     2
## 202     8     2
## 325     8     2
## 793     8     2
## 847     8     2
## 864     8     2
```

Ticket 347088:

```
select(ds[ds$Ticket == "347088", ], Name, Pclass, Fare, Cabin, Embarked, Sex, Age, SibSp, Parch)
```

```
##                                                Name Pclass Fare Cabin
## 64                            Skoog, Master. Harald      3 27.9
## 168 Skoog, Mrs. William (Anna Bernhardina Karlsson)      3 27.9
## 361                              Skoog, Mr. Wilhelm      3 27.9
## 635                              Skoog, Miss. Mabel      3 27.9
## 643                   Skoog, Miss. Margit Elizabeth      3 27.9
## 820                    Skoog, Master. Karl Thorsten      3 27.9
##        Embarked    Sex Age SibSp Parch
## 64  Southampton   male   4     3     2
## 168 Southampton female  45     1     4
## 361 Southampton   male  40     1     4
## 635 Southampton female   9     3     2
## 643 Southampton female   2     3     2
## 820 Southampton   male  10     3     2
```

Ticket 3101295:

```
select(ds[ds$Ticket == "3101295", ], Name, Pclass, Fare, Cabin, Embarked, Sex, Age, SibSp, Parch)
```

```
##                                       Name Pclass    Fare Cabin    Embarked
## 51              Panula, Master. Juha Niilo      3 39.6875       Southampton
## 165           Panula, Master. Eino Viljami      3 39.6875       Southampton
## 267              Panula, Mr. Ernesti Arvid      3 39.6875       Southampton
## 639 Panula, Mrs. Juha (Maria Emilia Ojala)      3 39.6875       Southampton
## 687               Panula, Mr. Jaako Arnold      3 39.6875       Southampton
## 825           Panula, Master. Urho Abraham      3 39.6875       Southampton
##        Sex Age SibSp Parch
## 51    male   7     4     1
## 165   male   1     4     1
## 267   male  16     4     1
## 639 female  41     0     5
## 687   male  14     4     1
## 825   male   2     4     1
```

Ticket 347088:

```
select(ds[ds$Ticket == "347088", ], Name, Pclass, Fare, Cabin, Embarked, Sex, Age, SibSp, Parch)
```

```
##                                                Name Pclass Fare Cabin
## 64                            Skoog, Master. Harald      3 27.9
## 168 Skoog, Mrs. William (Anna Bernhardina Karlsson)      3 27.9
## 361                              Skoog, Mr. Wilhelm      3 27.9
## 635                              Skoog, Miss. Mabel      3 27.9
## 643                   Skoog, Miss. Margit Elizabeth      3 27.9
## 820                    Skoog, Master. Karl Thorsten      3 27.9
##        Embarked    Sex Age SibSp Parch
## 64  Southampton   male   4     3     2
## 168 Southampton female  45     1     4
## 361 Southampton   male  40     1     4
## 635 Southampton female   9     3     2
## 643 Southampton female   2     3     2
## 820 Southampton   male  10     3     2
```

Ticket CA 2144:

```r
select(ds[ds$Ticket == "CA 2144", ], Name, Pclass, Fare, Cabin, Embarked, Sex, Age, SibSp, Parch)
```

```
##                                         Name Pclass Fare Cabin    Embarked
## 60       Goodwin, Master. William Frederick      3 46.9       Southampton
## 72               Goodwin, Miss. Lillian Amy      3 46.9       Southampton
## 387         Goodwin, Master. Sidney Leonard      3 46.9       Southampton
## 481          Goodwin, Master. Harold Victor      3 46.9       Southampton
## 679 Goodwin, Mrs. Frederick (Augusta Tyler)      3 46.9       Southampton
## 684             Goodwin, Mr. Charles Edward      3 46.9       Southampton
##        Sex Age SibSp Parch
## 60    male  11     5     2
## 72  female  16     5     2
## 387   male   1     5     2
## 481   male   9     5     2
## 679 female  43     1     6
## 684   male  14     5     2
```

Ticket 382652:

```r
select(ds[ds$Ticket == "382652", ], Name, Pclass, Fare, Cabin, Embarked, Sex, Age, SibSp, Parch)
```

```
##                                      Name Pclass   Fare Cabin   Embarked    Sex
## 17                  Rice, Master. Eugene      3 29.125       Queenstown   male
## 172                 Rice, Master. Arthur      3 29.125       Queenstown   male
## 279                   Rice, Master. Eric      3 29.125       Queenstown   male
## 788            Rice, Master. George Hugh      3 29.125       Queenstown   male
## 886 Rice, Mrs. William (Margaret Norton)      3 29.125       Queenstown female
##     Age SibSp Parch
## 17    2     4     1
## 172   4     4     1
## 279   7     4     1
## 788   8     4     1
## 886  39     0     5
```

Ticket S.O.C. 14879:

```r
select(ds[ds$Ticket == "S.O.C. 14879", ], Name, Pclass, Fare, Cabin, Embarked, Sex, Age, SibSp, Parch)
```

```
##                             Name Pclass Fare Cabin    Embarked  Sex Age SibSp
## 73         Hood, Mr. Ambrose Jr      2 73.5       Southampton male  21     0
## 121 Hickman, Mr. Stanley George      2 73.5       Southampton male  21     2
## 386   Davies, Mr. Charles Henry      2 73.5       Southampton male  18     0
## 656   Hickman, Mr. Leonard Mark      2 73.5       Southampton male  24     2
## 666          Hickman, Mr. Lewis      2 73.5       Southampton male  32     2
##     Parch
## 73      0
## 121     0
## 386     0
## 656     0
## 666     0
```

```r
sort(distinct(ds, Ticket)$Ticket)
```

```
##   [1] "110152"             "110413"             "110465"
##   [4] "110564"             "110813"             "111240"
##   [7] "111320"             "111361"             "111369"
##  [10] "111426"             "111427"             "111428"
##  [13] "112050"             "112052"             "112053"
##  [16] "112058"             "112059"             "112277"
##  [19] "112379"             "113028"             "113043"
##  [22] "113050"             "113051"             "113055"
##  [25] "113056"             "113059"             "113501"
##  [28] "113503"             "113505"             "113509"
##  [31] "113510"             "113514"             "113572"
##  [34] "113760"             "113767"             "113773"
##  [37] "113776"             "113781"             "113783"
##  [40] "113784"             "113786"             "113787"
##  [43] "113788"             "113789"             "113792"
##  [46] "113794"             "113796"             "113798"
##  [49] "113800"             "113803"             "113804"
##  [52] "113806"             "113807"             "11668"
##  [55] "11751"              "11752"              "11753"
##  [58] "11755"              "11765"              "11767"
##  [61] "11769"              "11771"              "11774"
##  [64] "11813"              "11967"              "12233"
##  [67] "12460"              "12749"              "13049"
##  [70] "13213"              "13214"              "13502"
##  [73] "13507"              "13509"              "13567"
##  [76] "13568"              "14311"              "14312"
##  [79] "14313"              "14973"              "1601"
##  [82] "16966"              "16988"              "17421"
##  [85] "17453"              "17463"              "17464"
##  [88] "17465"              "17466"              "17474"
##  [91] "17764"              "19877"              "19928"
##  [94] "19943"              "19947"              "19950"
##  [97] "19952"              "19972"              "19988"
## [100] "19996"              "2003"               "211536"
## [103] "21440"              "218629"             "219533"
## [106] "220367"             "220845"             "2223"
## [109] "223596"             "226593"             "226875"
## [112] "228414"             "229236"             "230080"
## [115] "230136"             "230433"             "230434"
## [118] "231919"             "231945"             "233639"
## [121] "233866"             "234360"             "234604"
## [124] "234686"             "234818"             "236171"
## [127] "236852"             "236853"             "237442"
## [130] "237565"             "237668"             "237671"
## [133] "237736"             "237789"             "237798"
## [136] "239853"             "239854"             "239855"
## [139] "239856"             "239865"             "240929"
## [142] "24160"              "243847"             "243880"
## [145] "244252"             "244270"             "244278"
## [148] "244310"             "244358"             "244361"
## [151] "244367"             "244373"             "248698"
## [154] "248706"             "248723"             "248727"
## [157] "248731"             "248733"             "248738"
## [160] "248740"             "248747"             "250643"
## [163] "250644"             "250646"             "250647"
## [166] "250648"             "250649"             "250651"
## [169] "250652"             "250653"             "250655"
## [172] "2620"               "2623"               "2624"
## [175] "2625"               "2626"               "2627"
## [178] "2628"               "2629"               "2631"
## [181] "26360"              "2641"               "2647"
## [184] "2648"               "2649"               "2650"
## [187] "2651"               "2653"               "2659"
## [190] "2661"               "2662"               "2663"
## [193] "2664"               "2665"               "2666"
## [196] "2667"               "2668"               "2669"
## [199] "26707"              "2671"               "2672"
## [202] "2674"               "2677"               "2678"
## [205] "2680"               "2683"               "2685"
## [208] "2686"               "2687"               "2689"
## [211] "2690"               "2691"               "2693"
## [214] "2694"               "2695"               "2697"
## [217] "2699"               "2700"               "27042"
## [220] "27267"              "27849"              "28134"
## [223] "28206"              "28213"              "28220"
## [226] "28228"              "28403"              "28424"
## [229] "28425"              "28551"              "28664"
## [232] "28665"              "29011"              "2908"
## [235] "29103"              "29104"              "29105"
## [238] "29106"              "29108"              "2926"
## [241] "29750"              "29751"              "3101264"
## [244] "3101265"            "3101267"            "3101276"
## [247] "3101277"            "3101278"            "3101281"
## [250] "3101295"            "3101296"            "3101298"
## [253] "31027"              "31028"              "312991"
## [256] "312992"             "312993"             "31418"
## [259] "315037"             "315082"             "315084"
## [262] "315086"             "315088"             "315089"
## [265] "315090"             "315093"             "315094"
## [268] "315096"             "315097"             "315098"
## [271] "315151"             "315153"             "323592"
## [274] "323951"             "324669"             "330877"
## [277] "330909"             "330919"             "330923"
## [280] "330931"             "330932"             "330935"
## [283] "330958"             "330959"             "330979"
## [286] "330980"             "334912"             "335097"
## [289] "335677"             "33638"              "336439"
## [292] "3411"               "341826"             "34218"
## [295] "342826"             "343095"             "343120"
## [298] "343275"             "343276"             "345364"
## [301] "345572"             "345763"             "345764"
## [304] "345765"             "345767"             "345769"
## [307] "345770"             "345773"             "345774"
## [310] "345777"             "345778"             "345779"
## [313] "345780"             "345781"             "345783"
## [316] "3460"               "347054"             "347060"
## [319] "347061"             "347062"             "347063"
## [322] "347064"             "347067"             "347068"
## [325] "347069"             "347071"             "347073"
## [328] "347074"             "347076"             "347077"
## [331] "347078"             "347080"             "347081"
## [334] "347082"             "347083"             "347085"
## [337] "347087"             "347088"             "347089"
## [340] "3474"               "347464"             "347466"
## [343] "347468"             "347470"             "347742"
## [346] "347743"             "348121"             "348123"
## [349] "348124"             "349201"             "349203"
## [352] "349204"             "349205"             "349206"
## [355] "349207"             "349208"             "349209"
## [358] "349210"             "349212"             "349213"
## [361] "349214"             "349215"             "349216"
## [364] "349217"             "349218"             "349219"
## [367] "349221"             "349222"             "349223"
## [370] "349224"             "349225"             "349227"
## [373] "349228"             "349231"             "349233"
## [376] "349234"             "349236"             "349237"
## [379] "349239"             "349240"             "349241"
## [382] "349242"             "349243"             "349244"
## [385] "349245"             "349246"             "349247"
## [388] "349248"             "349249"             "349251"
## [391] "349252"             "349253"             "349254"
## [394] "349256"             "349257"             "349909"
## [397] "349910"             "349912"             "350025"
## [400] "350026"             "350029"             "350034"
## [403] "350035"             "350036"             "350042"
## [406] "350043"             "350046"             "350047"
## [409] "350048"             "350050"             "350052"
## [412] "350060"             "350404"             "350406"
## [415] "350407"             "350417"             "35273"
## [418] "35281"              "35851"              "35852"
## [421] "358585"             "36209"              "362316"
## [424] "363291"             "363294"             "363592"
## [427] "364498"             "364499"             "364500"
## [430] "364506"             "364511"             "364512"
## [433] "364516"             "364846"             "364848"
## [436] "364849"             "364850"             "364851"
## [439] "365222"             "365226"             "36568"
## [442] "367226"             "367228"             "367229"
## [445] "367230"             "367231"             "367232"
## [448] "367655"             "368323"             "36864"
## [451] "36865"              "36866"              "368703"
## [454] "36928"              "36947"              "36963"
## [457] "36967"              "36973"              "370129"
## [460] "370365"             "370369"             "370370"
## [463] "370371"             "370372"             "370373"
## [466] "370375"             "370376"             "370377"
## [469] "371060"             "371110"             "371362"
## [472] "372622"             "373450"             "374746"
## [475] "374887"             "374910"             "376564"
## [478] "376566"             "382649"             "382651"
## [481] "382652"             "383121"             "384461"
## [484] "386525"             "392091"             "392092"
## [487] "392096"             "394140"             "4133"
## [490] "4134"               "4135"               "4136"
## [493] "4137"               "4138"               "4579"
## [496] "54636"              "5727"               "65303"
## [499] "65304"              "65306"              "6563"
## [502] "693"                "695"                "7267"
## [505] "7534"               "7540"               "7545"
## [508] "7546"               "7552"               "7553"
## [511] "7598"               "8471"               "8475"
## [514] "9234"               "A./5. 2152"         "A./5. 3235"
## [517] "A.5. 11206"         "A.5. 18509"         "A/4 45380"
## [520] "A/4 48871"          "A/4. 20589"         "A/4. 34244"
## [523] "A/4. 39886"         "A/5 21171"          "A/5 21172"
## [526] "A/5 21173"          "A/5 21174"          "A/5 2466"
## [529] "A/5 2817"           "A/5 3536"           "A/5 3540"
## [532] "A/5 3594"           "A/5 3902"           "A/5. 10482"
## [535] "A/5. 13032"         "A/5. 2151"          "A/5. 3336"
## [538] "A/5. 3337"          "A/5. 851"           "A/S 2816"
## [541] "A4. 54510"          "C 17369"            "C 4001"
## [544] "C 7075"             "C 7076"             "C 7077"
## [547] "C.A. 17248"         "C.A. 18723"         "C.A. 2315"
## [550] "C.A. 24579"         "C.A. 24580"         "C.A. 2673"
## [553] "C.A. 29178"         "C.A. 29395"         "C.A. 29566"
## [556] "C.A. 31026"         "C.A. 31921"         "C.A. 33111"
## [559] "C.A. 33112"         "C.A. 33595"         "C.A. 34260"
## [562] "C.A. 34651"         "C.A. 37671"         "C.A. 5547"
## [565] "C.A. 6212"          "C.A./SOTON 34068"   "CA 2144"
## [568] "CA. 2314"           "CA. 2343"           "F.C. 12750"
## [571] "F.C.C. 13528"       "F.C.C. 13529"       "F.C.C. 13531"
## [574] "Fa 265302"          "LINE"               "P/PP 3381"
## [577] "PC 17318"           "PC 17473"           "PC 17474"
## [580] "PC 17475"           "PC 17476"           "PC 17477"
## [583] "PC 17482"           "PC 17483"           "PC 17485"
## [586] "PC 17558"           "PC 17569"           "PC 17572"
## [589] "PC 17582"           "PC 17585"           "PC 17590"
## [592] "PC 17592"           "PC 17593"           "PC 17595"
## [595] "PC 17596"           "PC 17597"           "PC 17599"
## [598] "PC 17600"           "PC 17601"           "PC 17603"
## [601] "PC 17604"           "PC 17605"           "PC 17608"
## [604] "PC 17609"           "PC 17610"           "PC 17611"
## [607] "PC 17612"           "PC 17754"           "PC 17755"
## [610] "PC 17756"           "PC 17757"           "PC 17758"
## [613] "PC 17759"           "PC 17760"           "PC 17761"
## [616] "PP 4348"            "PP 9549"            "S.C./A.4. 23567"
## [619] "S.C./PARIS 2079"    "S.O./P.P. 3"        "S.O./P.P. 751"
## [622] "S.O.C. 14879"       "S.O.P. 1166"        "S.P. 3464"
## [625] "S.W./PP 752"        "SC 1748"            "SC/AH 29037"
## [628] "SC/AH 3085"         "SC/AH Basle 541"    "SC/Paris 2123"
## [631] "SC/PARIS 2131"      "SC/PARIS 2133"      "SC/PARIS 2146"
## [634] "SC/PARIS 2149"      "SC/Paris 2163"      "SC/PARIS 2167"
## [637] "SCO/W 1585"         "SO/C 14885"         "SOTON/O.Q. 3101305"
## [640] "SOTON/O.Q. 3101306" "SOTON/O.Q. 3101307" "SOTON/O.Q. 3101310"
## [643] "SOTON/O.Q. 3101311" "SOTON/O.Q. 3101312" "SOTON/O.Q. 392078"
## [646] "SOTON/O.Q. 392087"  "SOTON/O2 3101272"   "SOTON/O2 3101287"
## [649] "SOTON/OQ 3101316"   "SOTON/OQ 3101317"   "SOTON/OQ 392076"
## [652] "SOTON/OQ 392082"    "SOTON/OQ 392086"    "SOTON/OQ 392089"
## [655] "SOTON/OQ 392090"    "STON/O 2. 3101269"  "STON/O 2. 3101273"
## [658] "STON/O 2. 3101274"  "STON/O 2. 3101275"  "STON/O 2. 3101280"
## [661] "STON/O 2. 3101285"  "STON/O 2. 3101286"  "STON/O 2. 3101288"
## [664] "STON/O 2. 3101289"  "STON/O 2. 3101292"  "STON/O 2. 3101293"
## [667] "STON/O 2. 3101294"  "STON/O2. 3101271"   "STON/O2. 3101279"
## [670] "STON/O2. 3101282"   "STON/O2. 3101283"   "STON/O2. 3101290"
## [673] "SW/PP 751"          "W./C. 14258"        "W./C. 14263"
## [676] "W./C. 6607"         "W./C. 6608"         "W./C. 6609"
## [679] "W.E.P. 5734"        "W/C 14208"          "WE/P 5735"
```

2.2 Selección y creación de características

Los atributos PassengerId y Name no serán objeto de análisis.

Nótese que Cabin es susceptible de ser dividida en letra y número.

2.1 Carga de los datos y selección

2.2 Transformación de los datos

A continuación analizamos cada uno de los distintos atributos:

```
summary(ds)
```

```
##   PassengerId    Survived      Pclass          Name               Sex
##  Min.   :  1.0   Not:549   Min.   :1.000   Length:891         female:314
##  1st Qu.:223.5   Yes:342   1st Qu.:2.000   Class :character   male  :577
##  Median :446.0             Median :3.000   Mode  :character
##  Mean   :446.0             Mean   :2.309
##  3rd Qu.:668.5             3rd Qu.:3.000
##  Max.   :891.0             Max.   :3.000
##
##       Age            SibSp           Parch            Ticket
##  Min.   : 0.42   Min.   :0.000   Min.   :0.0000   Length:891
##  1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000   Class :character
##  Median :28.00   Median :0.000   Median :0.0000   Mode  :character
##  Mean   :29.70   Mean   :0.523   Mean   :0.3816
##  3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
##  Max.   :80.00   Max.   :8.000   Max.   :6.0000
##  NA's   :177
##       Fare           Cabin                  Embarked
##  Min.   :  0.00   Length:891         Cherbourg  :168
##  1st Qu.:  7.91   Class :character   Queenstown : 77
##  Median : 14.45   Mode  :character   Southampton:644
##  Mean   : 32.20                      NA's       :  2
##  3rd Qu.: 31.00
##  Max.   :512.33
##
```

Vemos que los campos Age y Embarked tienen 177 y 2 valores nulos, respectivamente. Como no tiene sentido interpretarlos como 0 años o ningún puerto, sustituimos estos campos por la mediana para que afecten en la medida de lo posible al análisis.

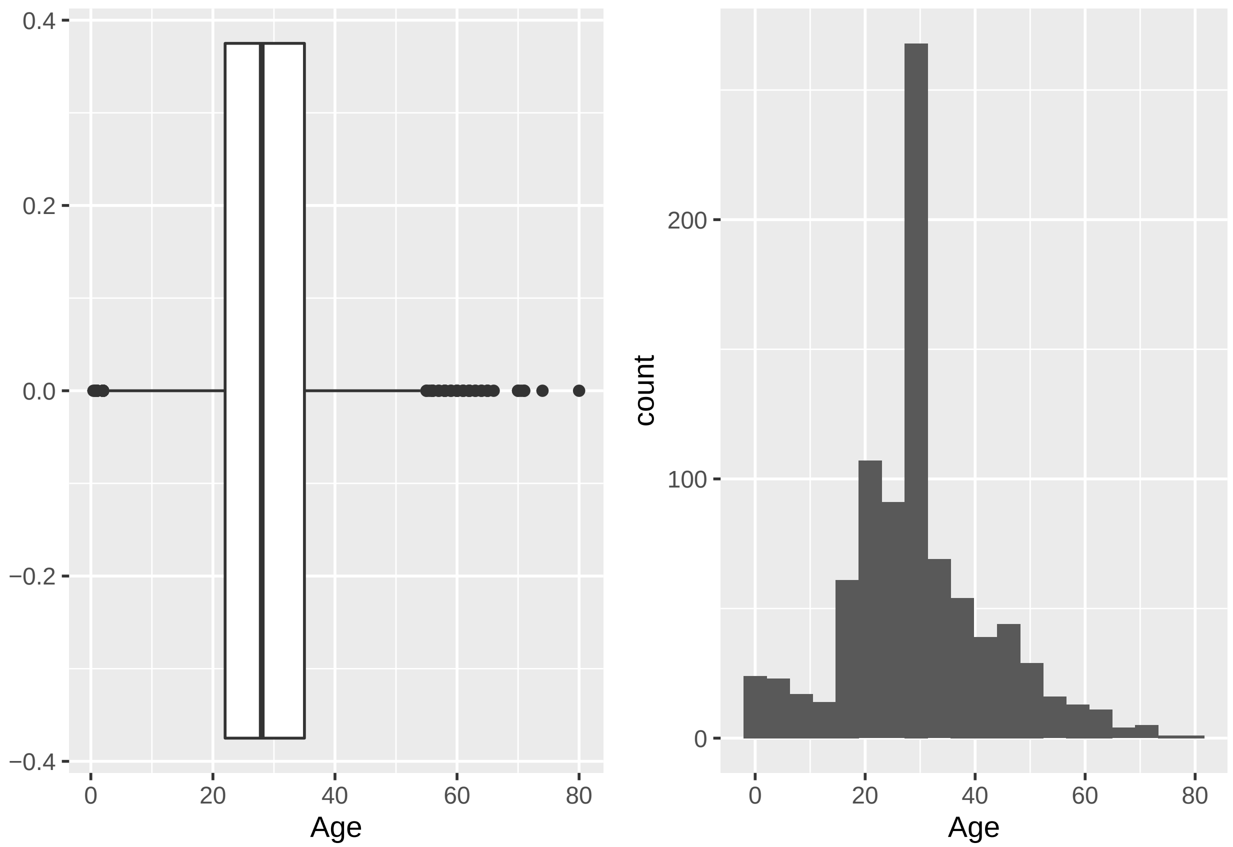
```
age_median <- median(ds$Age, na.rm = TRUE)

ds[, 'Age'][is.na(ds[,'Age'])] <- age_median

embarked_most_frequent <-  levels(ds$Embarked)[which.max(ds$Embarked)]
```

```r
ds[, 'Embarked'][is.na(ds[,'Embarked'])] <- embarked_most_frequent

summary(ds)
```

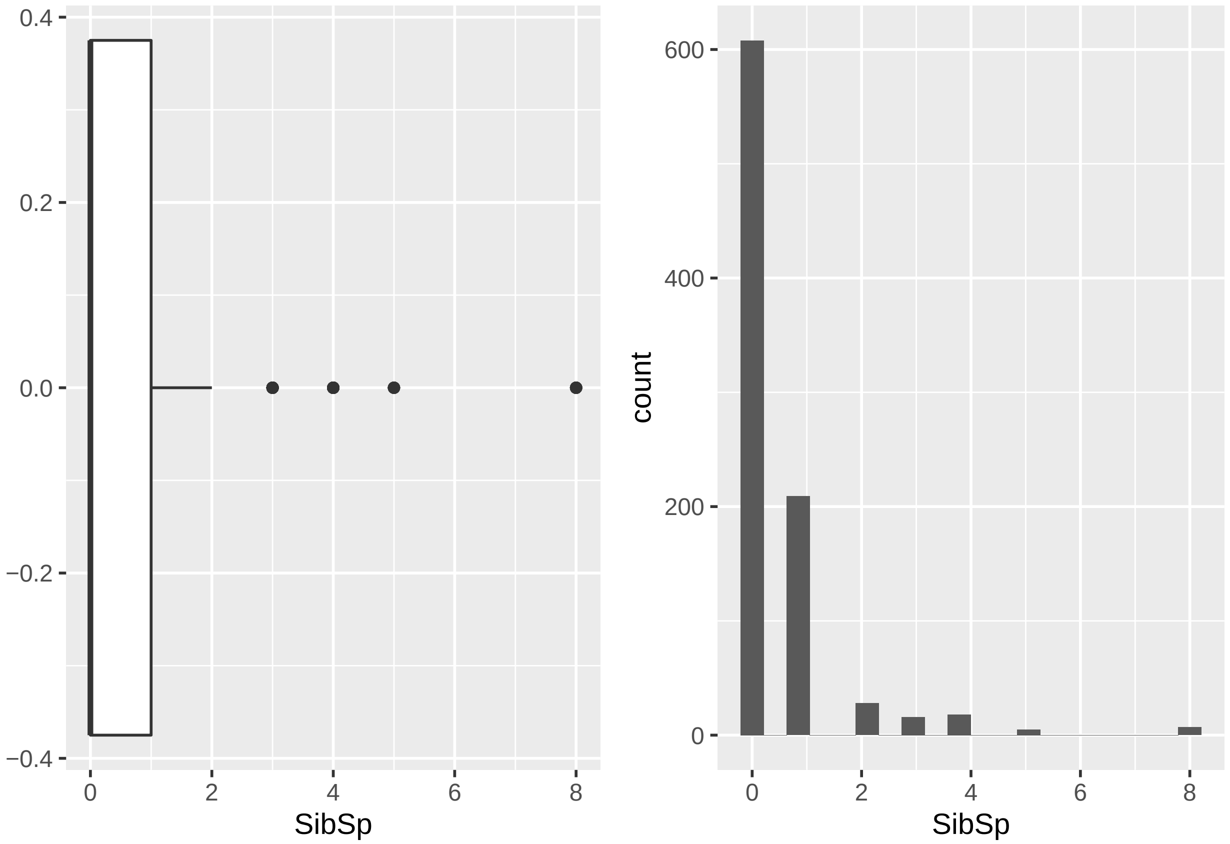
```
##   PassengerId    Survived      Pclass          Name                Sex
##  Min.   :  1.0   Not:549   Min.   :1.000   Length:891         female:314
##  1st Qu.:223.5   Yes:342   1st Qu.:2.000   Class :character   male  :577
##  Median :446.0             Median :3.000   Mode  :character
##  Mean   :446.0             Mean   :2.309
##  3rd Qu.:668.5             3rd Qu.:3.000
##  Max.   :891.0             Max.   :3.000
##       Age            SibSp           Parch           Ticket
##  Min.   : 0.42   Min.   :0.000   Min.   :0.0000   Length:891
##  1st Qu.:22.00   1st Qu.:0.000   1st Qu.:0.0000   Class :character
##  Median :28.00   Median :0.000   Median :0.0000   Mode  :character
##  Mean   :29.36   Mean   :0.523   Mean   :0.3816
##  3rd Qu.:35.00   3rd Qu.:1.000   3rd Qu.:0.0000
##  Max.   :80.00   Max.   :8.000   Max.   :6.0000
##       Fare            Cabin                  Embarked
##  Min.   :  0.00   Length:891         Cherbourg  :170
##  1st Qu.:  7.91   Class :character   Queenstown : 77
##  Median : 14.45   Mode  :character   Southampton:644
##  Mean   : 32.20
##  3rd Qu.: 31.00
##  Max.   :512.33
```

```r
#Visualización de variables cuantitativas

#Age
gAge1 <- ggplot(ds, aes(x=Age)) + geom_boxplot()

gAge2 <- ggplot(ds, aes(x=Age)) + geom_histogram(bins=20)

#SibSp

gSibSp1 <- ggplot(ds, aes(x=SibSp)) + geom_boxplot()

gSibSp2 <- ggplot(ds, aes(x=SibSp)) + geom_histogram(bins=20)

#Parch

gParch1 <- ggplot(ds, aes(x=Parch)) + geom_boxplot()

gParch2 <- ggplot(ds, aes(x=Parch)) + geom_histogram(bins=20)


#Fare

gFare1 <- ggplot(ds, aes(x=Fare)) + geom_boxplot()

gFare2 <- ggplot(ds, aes(x=Fare)) + geom_histogram(bins=20)
```

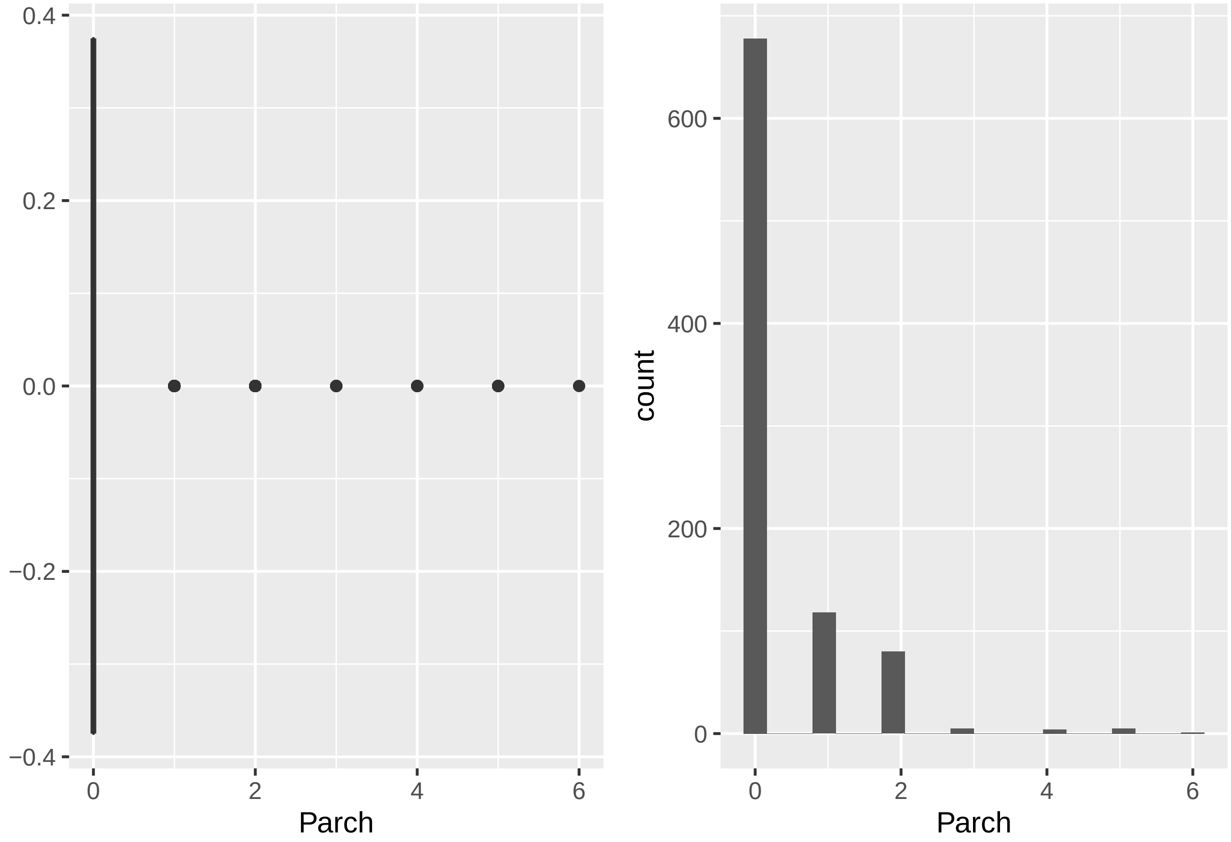
```r
grid.arrange(gAge1,gAge2,nrow=1)
```



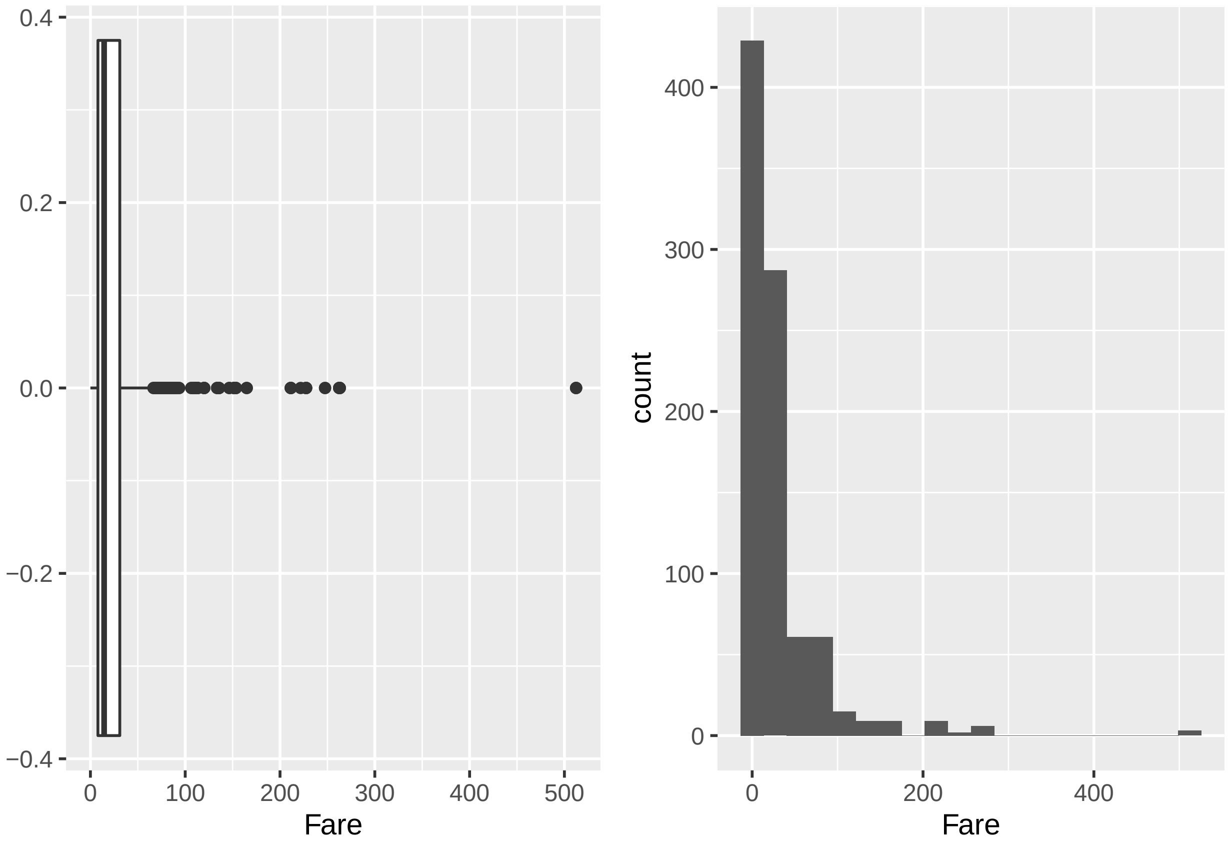
```r
grid.arrange(gSibSp1,gSibSp2,nrow=1)
```



```r
grid.arrange(gParch1,gParch2,nrow=1)
```



```r
grid.arrange(gFare1,gFare2,nrow=1)
```



```
#Visualizacion de variables cuantitativas

#Survived
sumSurvived <- summarize( group_by(ds, Survived), n=length(Survived), Fare=mean(Fare))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
gSurvived1 <- ggplot( sumSurvived, aes(x="", y=n, fill=Survived)) +
geom_bar(width = 1, stat = "identity") +
coord_polar("y", start=0) + ggtitle("Survived")

#PClass and Survived
sumPClass <- summarize( group_by(ds, Pclass), n=length(Pclass), Survived=mean(Survived))
```

```
## Warning in mean.default(Survived): argument is not numeric or logical: returning
## NA

## Warning in mean.default(Survived): argument is not numeric or logical: returning
## NA

## Warning in mean.default(Survived): argument is not numeric or logical: returning
## NA

## `summarise()` ungrouping output (override with `.groups` argument)
```

```
gPClass1 <- ggplot( sumPClass, aes(x="", y=n, fill=Pclass)) +
geom_bar(width = 1, stat = "identity") +
coord_polar("y", start=0) + ggtitle("PClass")
```

```r
gPClass2 <- ds %>%
  group_by(Survived, Pclass) %>%
  tally() %>%
  group_by(Survived) %>%
  mutate(x = n / sum(n)) %>%
  ggplot() +
    geom_col(aes(
      x = factor(Survived),
      y = x,
      fill = factor(Pclass)
      ), position = "stack")

#Sex and Survived
sumSex <- summarize( group_by(ds, Sex), n=length(Sex), Survived=mean(Survived))
```

```
## Warning in mean.default(Survived): argument is not numeric or logical: returning
## NA

## Warning in mean.default(Survived): argument is not numeric or logical: returning
## NA
```

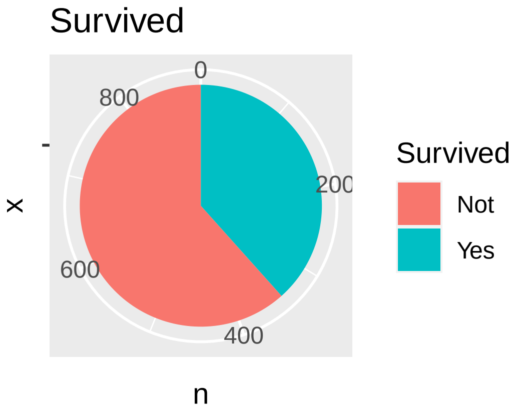
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```r
gSex1 <- ggplot( sumSex, aes(x="", y=n, fill=Sex)) +
geom_bar(width = 1, stat = "identity") +
coord_polar("y", start=0) + ggtitle("Sex")

gSex2 <- ds %>%
  group_by(Survived, Sex) %>%
  tally() %>%
  group_by(Survived) %>%
  mutate(x = n / sum(n)) %>%
  ggplot() +
    geom_col(aes(
      x = factor(Survived),
      y = x,
      fill = factor(Sex)
      ), position = "stack")

#Embarked and Survived
sumEmbarked <- summarize( group_by(ds, Embarked), n=length(Embarked))
```

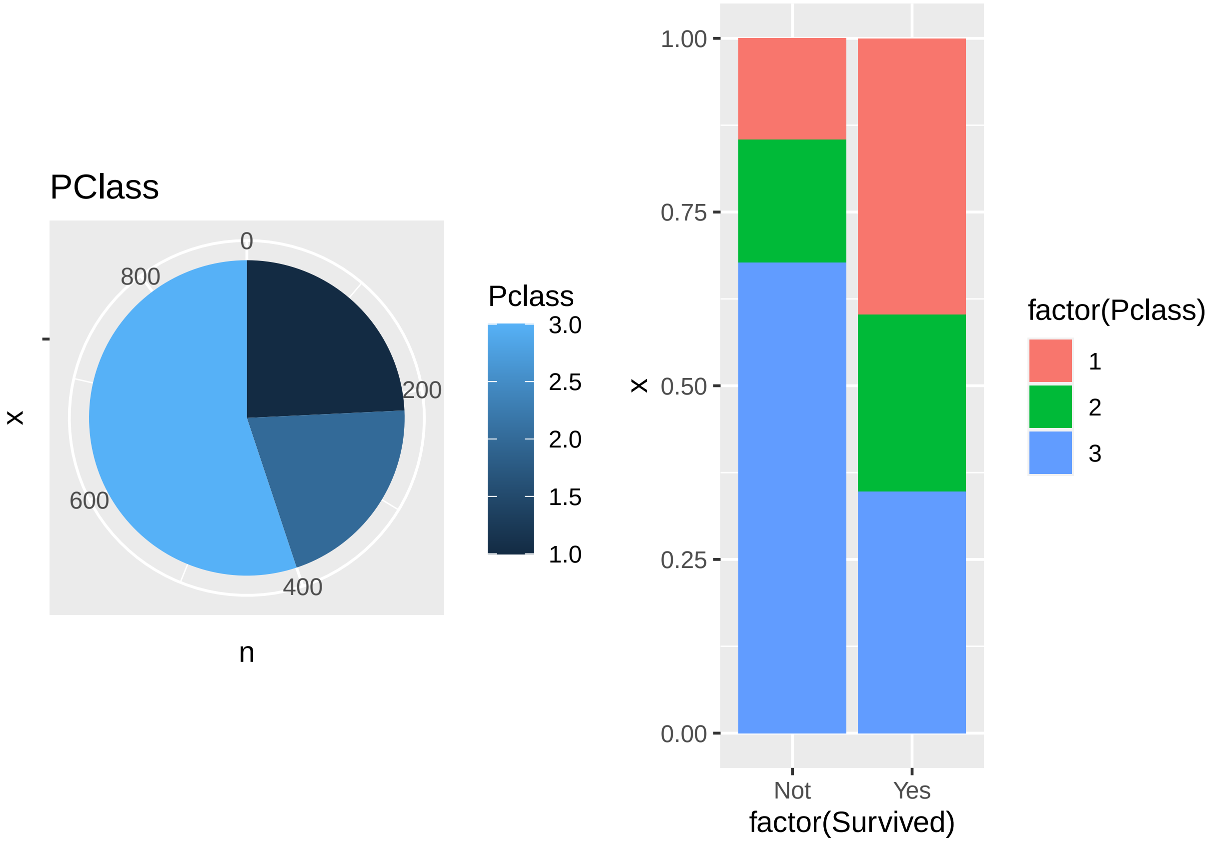
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```r
gEmbarked1 <- ggplot( sumEmbarked, aes(x="", y=n, fill=Embarked)) +
geom_bar(width = 1, stat = "identity") +
coord_polar("y", start=0) + ggtitle("Embarked")


gEmbarked2 <- ds %>%
  group_by(Survived, Embarked) %>%
  tally() %>%
  group_by(Survived) %>%
  mutate(x = n / sum(n)) %>%
  ggplot() +
    geom_col(aes(
      x = factor(Embarked),
      y = x,
      fill = factor(Survived)
      ), position = "stack")
```

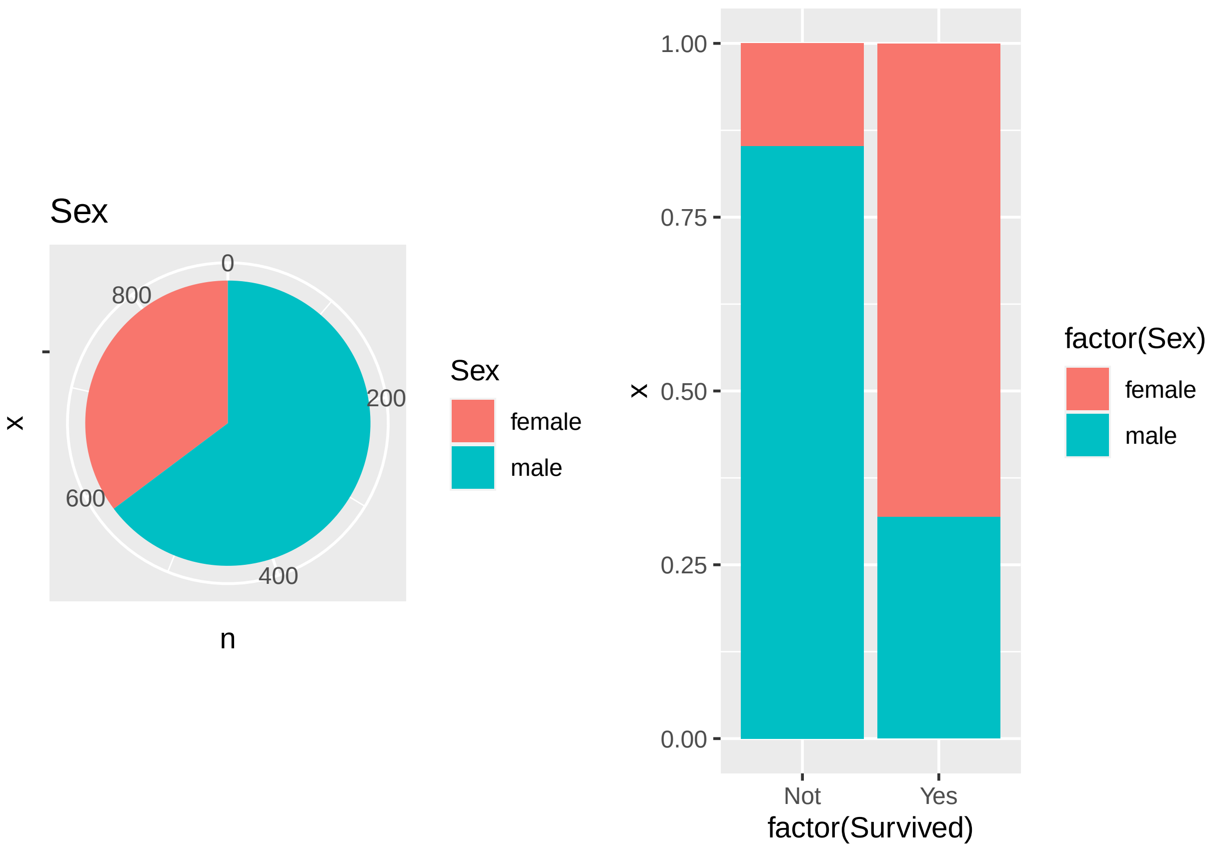
```
grid.arrange(gSurvived1, nrow=2)
```



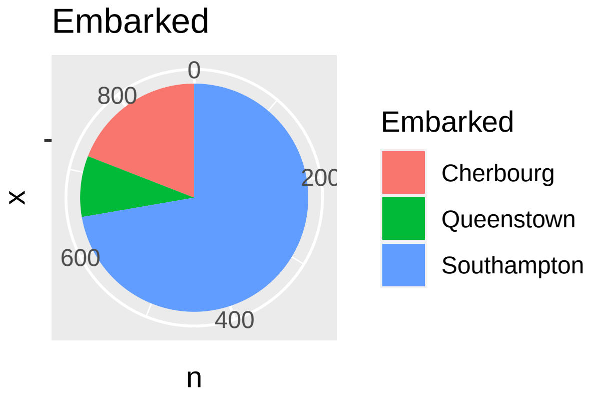
```
grid.arrange(gPClass1,gPClass2, nrow=1)
```

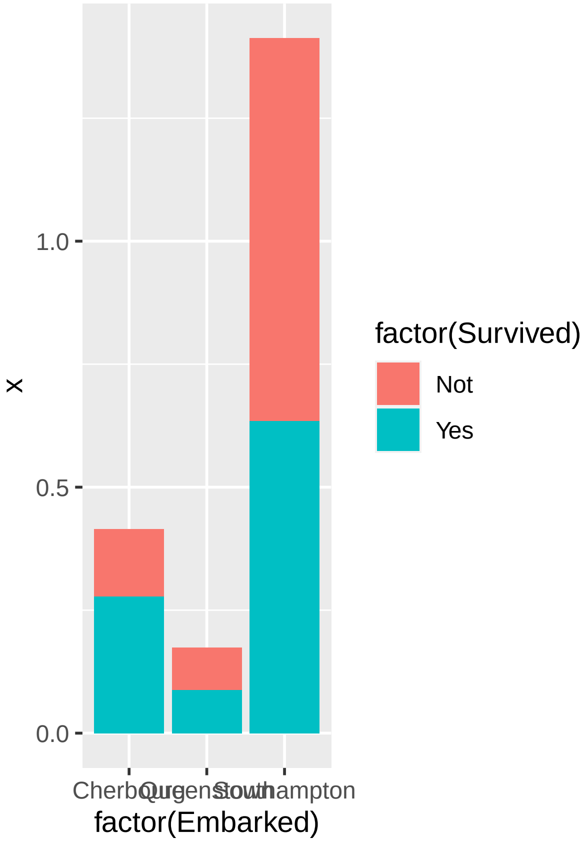


```
grid.arrange(gSex1, gSex2, nrow=1)
```



```
grid.arrange(gEmbarked1, gEmbarked2, nrow=1)
```





#2.3 Descipción estadística descriptiva

TODO: Describir cómo se distribuyen los datos y como podría saltar a la vista correlaciones. Da idea del ejercicio 4.

#3. Limpieza de datos

3.1 Elementos vacíos

TODO: En el ejercicio 1 se ha pintado el campo Age y el campo Embarked ya sin elementos vacíos. Traer aqui y pintar de nuevo, con un summary para demostrar que han desaparecido.

3.2 Identificación y tratamiento de valores extremos.

TODO: Explicar que hay valores extremos pero no podemos suponer que sean incorrectos (por ejemplo gente que tiene 8 hermanos o un billete que cuesta 500$). Poner ejemplos…

4. Análisis de los datos

Antes de proceder a ver qué grupos de datos queremos normalizar, vamos a ver qué datos son normales y cuáles no, de manera gráfica…

4.1 Selección de los grupos de datos que se quieen analizar / comparar.

Posibles grupos que puedan ser interesantes a la hora de averiguar si la probabilidad de sobrevivir era mucho más alta según el grupo.

```r
# Por clase

# Gente que viajaba sola vs gente con familia

# Por sexo

males_passengers <- ds[ds$Sex == "male",]
females_passengers <- ds[ds$Sex == "female",]


# Por edad, niños por debajo de los 16 años

children_passengers <- ds[ds$Age <= 16,]
not_children__passengers <- ds[ds$Age > 16,]
```

4.2. Comprobación de la normalidad y homogeneidad de la varianza

```r
#Normalidad para el campo Age
ks.test(ds$Age, pnorm, mean(ds$Age), sd(ds$Age))
```

```
## Warning in ks.test(ds$Age, pnorm, mean(ds$Age), sd(ds$Age)): ties should not be
## present for the Kolmogorov-Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  ds$Age
## D = 0.14658, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```r
shapiro.test(ds$Age)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  ds$Age
## W = 0.9541, p-value = 4.651e-16
```

Nos dan que la edad no sigue una distribucion normal, pero por el teorema del limite central. . . . podemos suponerlo.

```r
library(car)
```

```
## Warning: package 'car' was built under R version 4.0.3
```

```
## Loading required package: carData
```

```
## Warning: package 'carData' was built under R version 4.0.3
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```r
levtest<- function(x, y) {
  leveneTest(dv~gr, data = rbind(data.frame(dv=x, gr='gr1'),
                               data.frame(dv=y, gr='gr2')), center='mean')
}

flignertest<- function(x, y) {
  fligner.test(dv~gr, data = rbind(data.frame(dv=x, gr='gr1'),
                               data.frame(dv=y, gr='gr2')))
}

levtest(children_passengers$Age, not_children__passengers$Age)
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = "mean")
##        Df F value    Pr(>F)    
## group   1  22.594 2.333e-06 ***
##       889                      
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
flignertest(children_passengers$Age, not_children__passengers$Age)
```

```
## 
##  Fligner-Killeen test of homogeneity of variances
## 
## data:  dv by gr
## Fligner-Killeen:med chi-squared = 4.0025, df = 1, p-value = 0.04543
```

```
var.test(children_passengers$Age, not_children__passengers$Age)
```

```
## 
##  F test to compare two variances
## 
## data:  children_passengers$Age and not_children__passengers$Age
## F = 0.26025, num df = 99, denom df = 790, p-value = 6.71e-14
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.1967717 0.3563239
## sample estimates:
## ratio of variances 
##          0.2602506
```

#4.3. Aplicación de pruebas estadísticas para comparar los grupos de datos Ambos grupos no tienen la misma varianza.

```
  #Hay correlacion

  wilcox.test(Age ~ Survived, data = ds)
```

```
## 
##  Wilcoxon rank sum test with continuity correction
## 
## data:  Age by Survived
## W = 97979, p-value = 0.2697
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(Age ~ Survived, data = children_passengers)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Age by Survived
## W = 1578.5, p-value = 0.01782
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(Age ~ Survived, data = not_children__passengers)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Age by Survived
## W = 68680, p-value = 0.2343
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(children_passengers$Age, not_children__passengers$Age, alternative = "two.sided")
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  children_passengers$Age and not_children__passengers$Age
## W = 0, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(children_passengers$Age, not_children__passengers$Age, alternative = "greater")
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  children_passengers$Age and not_children__passengers$Age
## W = 0, p-value = 1
## alternative hypothesis: true location shift is greater than 0
```

```r
wilcox.test(children_passengers$Age, not_children__passengers$Age, alternative = "less")
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  children_passengers$Age and not_children__passengers$Age
## W = 0, p-value < 2.2e-16
## alternative hypothesis: true location shift is less than 0
```

```r
summary(children_passengers)
```

```
##   PassengerId     Survived     Pclass          Name                Sex
##  Min.   :  8.0   Not:45   Min.   :1.00   Length:100         female:49
##  1st Qu.:172.8   Yes:55   1st Qu.:2.00   Class :character   male   :51
##  Median :414.0            Median :3.00   Mode  :character
##  Mean   :430.0            Mean   :2.61
##  3rd Qu.:723.8            3rd Qu.:3.00
##  Max.   :876.0            Max.   :3.00
##       Age             SibSp          Parch          Ticket
##  Min.   : 0.420   Min.   :0.00   Min.   :0.00   Length:100
##  1st Qu.: 3.000   1st Qu.:0.00   1st Qu.:1.00   Class :character
##  Median : 7.500   Median :1.00   Median :1.00   Mode  :character
##  Mean   : 8.007   Mean   :1.57   Mean   :1.14
##  3rd Qu.:14.000   3rd Qu.:3.00   3rd Qu.:2.00
##  Max.   :16.000   Max.   :5.00   Max.   :3.00
##       Fare            Cabin                  Embarked
##  Min.   :  7.225   Length:100         Cherbourg  :19
##  1st Qu.: 14.489   Class :character   Queenstown : 7
##  Median : 26.000   Mode  :character   Southampton:74
##  Mean   : 31.589
##  3rd Qu.: 34.375
##  Max.   :211.338
```

```
  summary(not_children__passengers)
```

```
##   PassengerId    Survived      Pclass          Name                Sex
##  Min.   :  1.0   Not:504   Min.   :1.000   Length:791         female:265
##  1st Qu.:231.5   Yes:287   1st Qu.:1.000   Class :character   male  :526
##  Median :451.0             Median :3.000   Mode  :character
##  Mean   :448.0             Mean   :2.271
##  3rd Qu.:663.5             3rd Qu.:3.000
##  Max.   :891.0             Max.   :3.000
##       Age            SibSp            Parch           Ticket
##  Min.   :17.00   Min.   :0.0000   Min.   :0.0000   Length:791
##  1st Qu.:25.00   1st Qu.:0.0000   1st Qu.:0.0000   Class :character
##  Median :28.00   Median :0.0000   Median :0.0000   Mode  :character
##  Mean   :32.06   Mean   :0.3906   Mean   :0.2857
##  3rd Qu.:36.00   3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :80.00   Max.   :8.0000   Max.   :6.0000
##       Fare            Cabin                  Embarked
##  Min.   :  0.000   Length:791         Cherbourg  :151
##  1st Qu.:  7.896   Class :character   Queenstown : 70
##  Median : 13.000   Mode  :character   Southampton:570
##  Mean   : 32.282
##  3rd Qu.: 30.035
##  Max.   :512.329
```

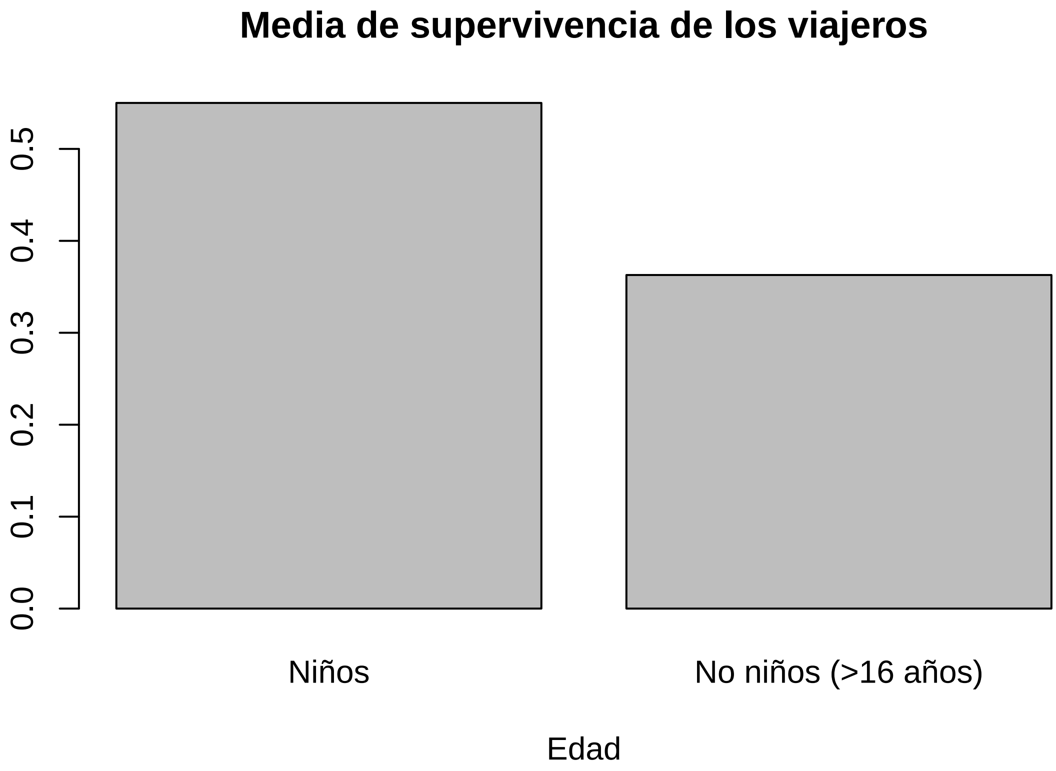
Los menores de 16 años tuvieron mucha más oportunidad de salvarse que los mayores de 16 años. La edad es un factor determinante. Pintarlo en el apartado 5.

5. Representación de los resultados a partir de tablas y gráficas

En el apartado anterior, hemos visto que los niños en particular y la edad en general han tenido un efecto importante sobre la supervivencia de los viajeros del Titanic.

Podemos ver que los menores de 16 años sobrevivieron mucho más que los mayores de 16 años:

```
children_passengers$Survived <-as.integer(children_passengers$Survived) - 1
not_children__passengers$Survived <- as.integer(not_children__passengers$Survived) - 1

mean_children_passengers <- mean(children_passengers$Survived)
mean_not_childre_passengers <- mean(not_children__passengers$Survived)

barplot(c(mean_children_passengers, mean_not_childre_passengers), names =c("Niños", "No niños (>16 años
```



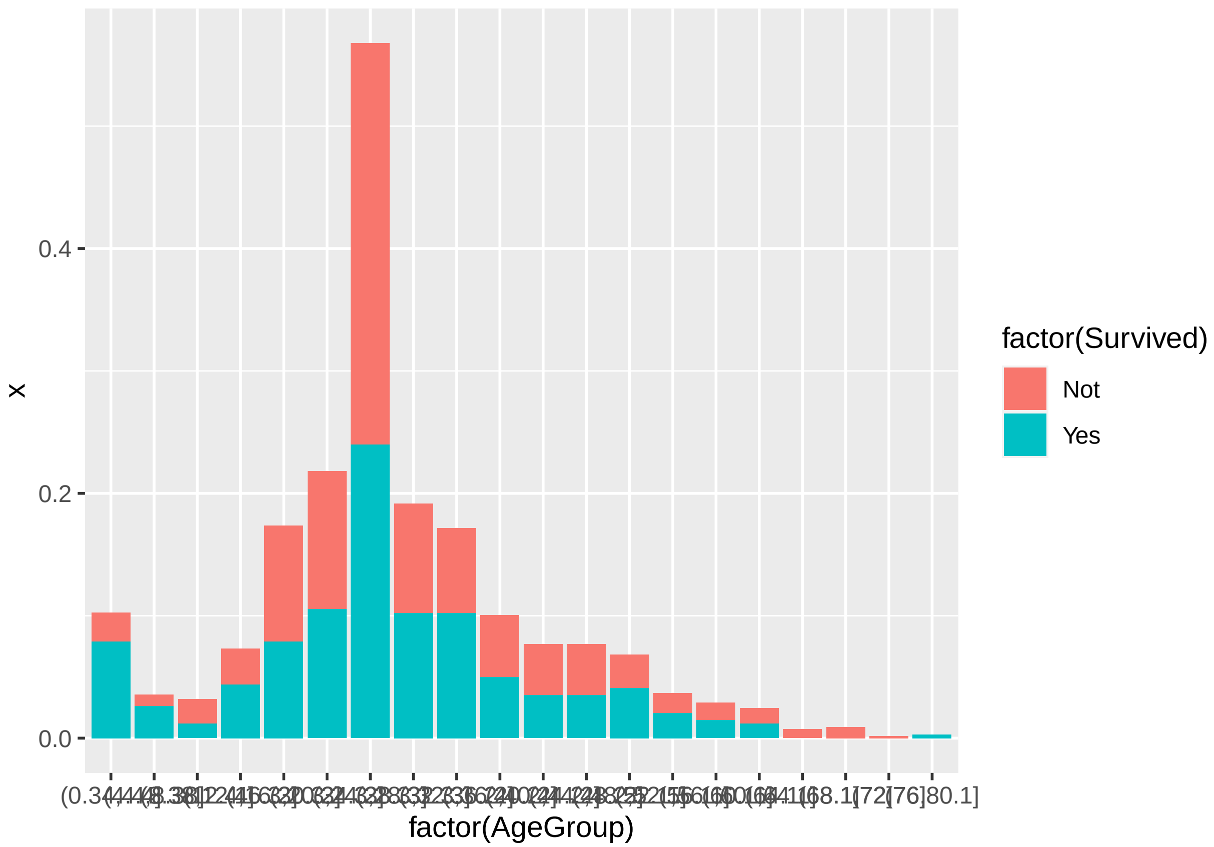
```
ds$AgeGroup <- cut(ds$Age, 20)

ds$AgeGroup
```

```
##   [1] (20.3,24.3] (36.2,40.2] (24.3,28.3] (32.3,36.2] (32.3,36.2] (24.3,28.3]
##   [7] (52.1,56.1] (0.34,4.4]  (24.3,28.3] (12.4,16.3] (0.34,4.4]  (56.1,60.1]
##  [13] (16.3,20.3] (36.2,40.2] (12.4,16.3] (52.1,56.1] (0.34,4.4]  (24.3,28.3]
##  [19] (28.3,32.3] (24.3,28.3] (32.3,36.2] (32.3,36.2] (12.4,16.3] (24.3,28.3]
##  [25] (4.4,8.38]  (36.2,40.2] (24.3,28.3] (16.3,20.3] (24.3,28.3] (24.3,28.3]
##  [31] (36.2,40.2] (24.3,28.3] (24.3,28.3] (64.1,68.1] (24.3,28.3] (40.2,44.2]
##  [37] (24.3,28.3] (20.3,24.3] (16.3,20.3] (12.4,16.3] (36.2,40.2] (24.3,28.3]
##  [43] (24.3,28.3] (0.34,4.4]  (16.3,20.3] (24.3,28.3] (24.3,28.3] (24.3,28.3]
##  [49] (24.3,28.3] (16.3,20.3] (4.4,8.38]  (20.3,24.3] (48.2,52.1] (28.3,32.3]
##  [55] (64.1,68.1] (24.3,28.3] (20.3,24.3] (28.3,32.3] (4.4,8.38]  (8.38,12.4]
##  [61] (20.3,24.3] (36.2,40.2] (44.2,48.2] (0.34,4.4]  (24.3,28.3] (24.3,28.3]
##  [67] (28.3,32.3] (16.3,20.3] (16.3,20.3] (24.3,28.3] (28.3,32.3] (12.4,16.3]
##  [73] (20.3,24.3] (24.3,28.3] (28.3,32.3] (24.3,28.3] (24.3,28.3] (24.3,28.3]
##  [79] (0.34,4.4]  (28.3,32.3] (20.3,24.3] (28.3,32.3] (24.3,28.3] (24.3,28.3]
##  [85] (16.3,20.3] (32.3,36.2] (12.4,16.3] (24.3,28.3] (20.3,24.3] (20.3,24.3]
##  [91] (28.3,32.3] (16.3,20.3] (44.2,48.2] (24.3,28.3] (56.1,60.1] (24.3,28.3]
##  [97] (68.1,72]   (20.3,24.3] (32.3,36.2] (32.3,36.2] (24.3,28.3] (24.3,28.3]
## [103] (20.3,24.3] (32.3,36.2] (36.2,40.2] (24.3,28.3] (20.3,24.3] (24.3,28.3]
## [109] (36.2,40.2] (24.3,28.3] (44.2,48.2] (12.4,16.3] (20.3,24.3] (16.3,20.3]
## [115] (16.3,20.3] (20.3,24.3] (68.1,72]   (28.3,32.3] (20.3,24.3] (0.34,4.4]
## [121] (20.3,24.3] (24.3,28.3] (32.3,36.2] (32.3,36.2] (52.1,56.1] (8.38,12.4]
## [127] (24.3,28.3] (20.3,24.3] (24.3,28.3] (44.2,48.2] (32.3,36.2] (16.3,20.3]
## [133] (44.2,48.2] (28.3,32.3] (24.3,28.3] (20.3,24.3] (16.3,20.3] (36.2,40.2]
## [139] (12.4,16.3] (20.3,24.3] (24.3,28.3] (20.3,24.3] (20.3,24.3] (16.3,20.3]
## [145] (16.3,20.3] (16.3,20.3] (24.3,28.3] (8.38,12.4] (36.2,40.2] (40.2,44.2]
## [151] (48.2,52.1] (20.3,24.3] (52.1,56.1] (40.2,44.2] (24.3,28.3] (48.2,52.1]
## [157] (12.4,16.3] (28.3,32.3] (24.3,28.3] (24.3,28.3] (40.2,44.2] (36.2,40.2]
## [163] (24.3,28.3] (16.3,20.3] (0.34,4.4]  (8.38,12.4] (24.3,28.3] (44.2,48.2]
## [169] (24.3,28.3] (24.3,28.3] (60.1,64.1] (0.34,4.4]  (0.34,4.4]  (20.3,24.3]
## [175] (52.1,56.1] (16.3,20.3] (24.3,28.3] (48.2,52.1] (28.3,32.3] (32.3,36.2]
## [181] (24.3,28.3] (24.3,28.3] (8.38,12.4] (0.34,4.4]  (0.34,4.4]  (24.3,28.3]
## [187] (24.3,28.3] (44.2,48.2] (36.2,40.2] (32.3,36.2] (28.3,32.3] (16.3,20.3]
## [193] (16.3,20.3] (0.34,4.4]  (40.2,44.2] (56.1,60.1] (24.3,28.3] (40.2,44.2]
## [199] (24.3,28.3] (20.3,24.3] (24.3,28.3] (24.3,28.3] (32.3,36.2] (44.2,48.2]
## [205] (16.3,20.3] (0.34,4.4]  (28.3,32.3] (24.3,28.3] (12.4,16.3] (36.2,40.2]
## [211] (20.3,24.3] (32.3,36.2] (20.3,24.3] (28.3,32.3] (24.3,28.3] (28.3,32.3]
## [217] (24.3,28.3] (40.2,44.2] (28.3,32.3] (28.3,32.3] (12.4,16.3] (24.3,28.3]
## [223] (48.2,52.1] (24.3,28.3] (36.2,40.2] (20.3,24.3] (16.3,20.3] (20.3,24.3]
## [229] (16.3,20.3] (24.3,28.3] (32.3,36.2] (28.3,32.3] (56.1,60.1] (4.4,8.38]
## [235] (20.3,24.3] (24.3,28.3] (40.2,44.2] (4.4,8.38]  (16.3,20.3] (32.3,36.2]
## [241] (24.3,28.3] (24.3,28.3] (28.3,32.3] (20.3,24.3] (28.3,32.3] (40.2,44.2]
## [247] (24.3,28.3] (20.3,24.3] (36.2,40.2] (52.1,56.1] (24.3,28.3] (28.3,32.3]
## [253] (60.1,64.1] (28.3,32.3] (40.2,44.2] (28.3,32.3] (24.3,28.3] (28.3,32.3]
## [259] (32.3,36.2] (48.2,52.1] (24.3,28.3] (0.34,4.4]  (48.2,52.1] (36.2,40.2]
## [265] (24.3,28.3] (32.3,36.2] (12.4,16.3] (24.3,28.3] (56.1,60.1] (32.3,36.2]
## [271] (24.3,28.3] (24.3,28.3] (40.2,44.2] (36.2,40.2] (24.3,28.3] (60.1,64.1]
## [277] (44.2,48.2] (24.3,28.3] (4.4,8.38]  (32.3,36.2] (64.1,68.1] (24.3,28.3]
## [283] (12.4,16.3] (16.3,20.3] (24.3,28.3] (32.3,36.2] (28.3,32.3] (20.3,24.3]
## [289] (40.2,44.2] (20.3,24.3] (24.3,28.3] (16.3,20.3] (32.3,36.2] (20.3,24.3]
## [295] (20.3,24.3] (24.3,28.3] (20.3,24.3] (0.34,4.4]  (24.3,28.3] (48.2,52.1]
## [301] (24.3,28.3] (24.3,28.3] (16.3,20.3] (24.3,28.3] (24.3,28.3] (0.34,4.4]
## [307] (24.3,28.3] (16.3,20.3] (28.3,32.3] (28.3,32.3] (20.3,24.3] (16.3,20.3]
## [313] (24.3,28.3] (24.3,28.3] (40.2,44.2] (24.3,28.3] (20.3,24.3] (52.1,56.1]
## [319] (28.3,32.3] (36.2,40.2] (20.3,24.3] (24.3,28.3] (28.3,32.3] (20.3,24.3]
## [325] (24.3,28.3] (32.3,36.2] (60.1,64.1] (32.3,36.2] (28.3,32.3] (12.4,16.3]
## [331] (24.3,28.3] (44.2,48.2] (36.2,40.2] (12.4,16.3] (24.3,28.3] (24.3,28.3]
## [337] (28.3,32.3] (40.2,44.2] (44.2,48.2] (44.2,48.2] (0.34,4.4]  (20.3,24.3]
## [343] (24.3,28.3] (24.3,28.3] (32.3,36.2] (20.3,24.3] (36.2,40.2] (24.3,28.3]
## [349] (0.34,4.4]  (40.2,44.2] (20.3,24.3] (24.3,28.3] (12.4,16.3] (24.3,28.3]
## [355] (24.3,28.3] (24.3,28.3] (20.3,24.3] (36.2,40.2] (24.3,28.3] (24.3,28.3]
## [361] (36.2,40.2] (28.3,32.3] (44.2,48.2] (32.3,36.2] (24.3,28.3] (28.3,32.3]
## [367] (56.1,60.1] (24.3,28.3] (24.3,28.3] (20.3,24.3] (24.3,28.3] (16.3,20.3]
## [373] (16.3,20.3] (20.3,24.3] (0.34,4.4]  (24.3,28.3] (20.3,24.3] (24.3,28.3]
## [379] (16.3,20.3] (16.3,20.3] (40.2,44.2] (0.34,4.4]  (28.3,32.3] (32.3,36.2]
## [385] (24.3,28.3] (16.3,20.3] (0.34,4.4]  (32.3,36.2] (24.3,28.3] (16.3,20.3]
## [391] (32.3,36.2] (20.3,24.3] (24.3,28.3] (20.3,24.3] (20.3,24.3] (20.3,24.3]
## [397] (28.3,32.3] (44.2,48.2] (20.3,24.3] (24.3,28.3] (36.2,40.2] (24.3,28.3]
## [403] (20.3,24.3] (24.3,28.3] (16.3,20.3] (32.3,36.2] (48.2,52.1] (0.34,4.4]
## [409] (20.3,24.3] (24.3,28.3] (24.3,28.3] (24.3,28.3] (32.3,36.2] (24.3,28.3]
## [415] (40.2,44.2] (24.3,28.3] (32.3,36.2] (16.3,20.3] (28.3,32.3] (8.38,12.4]
## [421] (24.3,28.3] (20.3,24.3] (28.3,32.3] (24.3,28.3] (16.3,20.3] (24.3,28.3]
## [427] (24.3,28.3] (16.3,20.3] (24.3,28.3] (28.3,32.3] (24.3,28.3] (24.3,28.3]
## [433] (40.2,44.2] (16.3,20.3] (48.2,52.1] (12.4,16.3] (20.3,24.3] (20.3,24.3]
## [439] (60.1,64.1] (28.3,32.3] (44.2,48.2] (16.3,20.3] (24.3,28.3] (24.3,28.3]
## [445] (24.3,28.3] (0.34,4.4]  (12.4,16.3] (32.3,36.2] (4.4,8.38]  (48.2,52.1]
## [451] (32.3,36.2] (24.3,28.3] (28.3,32.3] (48.2,52.1] (24.3,28.3] (28.3,32.3]
## [457] (64.1,68.1] (24.3,28.3] (48.2,52.1] (24.3,28.3] (44.2,48.2] (32.3,36.2]
## [463] (44.2,48.2] (44.2,48.2] (24.3,28.3] (36.2,40.2] (24.3,28.3] (52.1,56.1]
## [469] (24.3,28.3] (0.34,4.4]  (24.3,28.3] (36.2,40.2] (32.3,36.2] (20.3,24.3]
## [475] (20.3,24.3] (24.3,28.3] (32.3,36.2] (28.3,32.3] (20.3,24.3] (0.34,4.4]
## [481] (8.38,12.4] (24.3,28.3] (48.2,52.1] (60.1,64.1] (24.3,28.3] (24.3,28.3]
## [487] (32.3,36.2] (56.1,60.1] (28.3,32.3] (8.38,12.4] (24.3,28.3] (20.3,24.3]
## [493] (52.1,56.1] (68.1,72]   (20.3,24.3] (24.3,28.3] (52.1,56.1] (24.3,28.3]
## [499] (24.3,28.3] (20.3,24.3] (16.3,20.3] (20.3,24.3] (24.3,28.3] (36.2,40.2]
## [505] (12.4,16.3] (16.3,20.3] (32.3,36.2] (24.3,28.3] (24.3,28.3] (24.3,28.3]
## [511] (28.3,32.3] (24.3,28.3] (32.3,36.2] (52.1,56.1] (20.3,24.3] (44.2,48.2]
## [517] (32.3,36.2] (24.3,28.3] (32.3,36.2] (28.3,32.3] (28.3,32.3] (20.3,24.3]
## [523] (24.3,28.3] (40.2,44.2] (24.3,28.3] (40.2,44.2] (48.2,52.1] (24.3,28.3]
## [529] (36.2,40.2] (20.3,24.3] (0.34,4.4]  (24.3,28.3] (16.3,20.3] (24.3,28.3]
## [535] (28.3,32.3] (4.4,8.38]  (44.2,48.2] (28.3,32.3] (24.3,28.3] (20.3,24.3]
## [541] (32.3,36.2] (8.38,12.4] (8.38,12.4] (28.3,32.3] (48.2,52.1] (60.1,64.1]
## [547] (16.3,20.3] (24.3,28.3] (32.3,36.2] (4.4,8.38]  (16.3,20.3] (24.3,28.3]
## [553] (24.3,28.3] (20.3,24.3] (20.3,24.3] (60.1,64.1] (44.2,48.2] (24.3,28.3]
## [559] (36.2,40.2] (32.3,36.2] (24.3,28.3] (36.2,40.2] (24.3,28.3] (24.3,28.3]
## [565] (24.3,28.3] (20.3,24.3] (16.3,20.3] (28.3,32.3] (24.3,28.3] (28.3,32.3]
## [571] (60.1,64.1] (52.1,56.1] (32.3,36.2] (24.3,28.3] (12.4,16.3] (16.3,20.3]
## [577] (32.3,36.2] (36.2,40.2] (24.3,28.3] (28.3,32.3] (24.3,28.3] (36.2,40.2]
## [583] (52.1,56.1] (32.3,36.2] (24.3,28.3] (16.3,20.3] (44.2,48.2] (56.1,60.1]
## [589] (20.3,24.3] (24.3,28.3] (32.3,36.2] (48.2,52.1] (44.2,48.2] (24.3,28.3]
## [595] (36.2,40.2] (32.3,36.2] (24.3,28.3] (48.2,52.1] (24.3,28.3] (48.2,52.1]
## [601] (20.3,24.3] (24.3,28.3] (24.3,28.3] (40.2,44.2] (32.3,36.2] (32.3,36.2]
## [607] (28.3,32.3] (24.3,28.3] (20.3,24.3] (36.2,40.2] (36.2,40.2] (24.3,28.3]
## [613] (24.3,28.3] (24.3,28.3] (32.3,36.2] (20.3,24.3] (32.3,36.2] (24.3,28.3]
## [619] (0.34,4.4]  (24.3,28.3] (24.3,28.3] (40.2,44.2] (16.3,20.3] (20.3,24.3]
## [625] (20.3,24.3] (60.1,64.1] (56.1,60.1] (20.3,24.3] (24.3,28.3] (24.3,28.3]
## [631] (76,80.1]   (48.2,52.1] (28.3,32.3] (24.3,28.3] (8.38,12.4] (24.3,28.3]
## [637] (28.3,32.3] (28.3,32.3] (40.2,44.2] (24.3,28.3] (16.3,20.3] (20.3,24.3]
## [643] (0.34,4.4]  (24.3,28.3] (0.34,4.4]  (44.2,48.2] (16.3,20.3] (52.1,56.1]
## [649] (24.3,28.3] (20.3,24.3] (24.3,28.3] (16.3,20.3] (20.3,24.3] (24.3,28.3]
## [655] (16.3,20.3] (20.3,24.3] (24.3,28.3] (28.3,32.3] (20.3,24.3] (56.1,60.1]
## [661] (48.2,52.1] (36.2,40.2] (44.2,48.2] (32.3,36.2] (16.3,20.3] (28.3,32.3]
## [667] (24.3,28.3] (24.3,28.3] (40.2,44.2] (24.3,28.3] (36.2,40.2] (28.3,32.3]
## [673] (68.1,72]   (28.3,32.3] (24.3,28.3] (16.3,20.3] (24.3,28.3] (16.3,20.3]
## [679] (40.2,44.2] (32.3,36.2] (24.3,28.3] (24.3,28.3] (16.3,20.3] (12.4,16.3]
## [685] (56.1,60.1] (24.3,28.3] (12.4,16.3] (16.3,20.3] (16.3,20.3] (12.4,16.3]
## [691] (28.3,32.3] (0.34,4.4]  (24.3,28.3] (24.3,28.3] (56.1,60.1] (48.2,52.1]
## [697] (40.2,44.2] (24.3,28.3] (48.2,52.1] (40.2,44.2] (16.3,20.3] (32.3,36.2]
## [703] (16.3,20.3] (24.3,28.3] (24.3,28.3] (36.2,40.2] (44.2,48.2] (40.2,44.2]
## [709] (20.3,24.3] (24.3,28.3] (20.3,24.3] (24.3,28.3] (44.2,48.2] (28.3,32.3]
## [715] (48.2,52.1] (16.3,20.3] (36.2,40.2] (24.3,28.3] (24.3,28.3] (32.3,36.2]
## [721] (4.4,8.38]  (16.3,20.3] (32.3,36.2] (48.2,52.1] (24.3,28.3] (16.3,20.3]
## [727] (28.3,32.3] (24.3,28.3] (24.3,28.3] (24.3,28.3] (28.3,32.3] (8.38,12.4]
## [733] (24.3,28.3] (20.3,24.3] (20.3,24.3] (28.3,32.3] (44.2,48.2] (32.3,36.2]
## [739] (24.3,28.3] (24.3,28.3] (24.3,28.3] (32.3,36.2] (20.3,24.3] (20.3,24.3]
## [745] (28.3,32.3] (68.1,72]   (12.4,16.3] (28.3,32.3] (16.3,20.3] (28.3,32.3]
## [751] (0.34,4.4]  (4.4,8.38]  (32.3,36.2] (20.3,24.3] (44.2,48.2] (0.34,4.4]
## [757] (24.3,28.3] (16.3,20.3] (32.3,36.2] (32.3,36.2] (24.3,28.3] (40.2,44.2]
## [763] (16.3,20.3] (32.3,36.2] (12.4,16.3] (48.2,52.1] (24.3,28.3] (28.3,32.3]
## [769] (24.3,28.3] (28.3,32.3] (20.3,24.3] (44.2,48.2] (56.1,60.1] (24.3,28.3]
## [775] (52.1,56.1] (16.3,20.3] (24.3,28.3] (4.4,8.38]  (24.3,28.3] (40.2,44.2]
## [781] (12.4,16.3] (16.3,20.3] (28.3,32.3] (24.3,28.3] (24.3,28.3] (24.3,28.3]
## [787] (16.3,20.3] (4.4,8.38]  (0.34,4.4]  (44.2,48.2] (24.3,28.3] (12.4,16.3]
## [793] (24.3,28.3] (24.3,28.3] (24.3,28.3] (36.2,40.2] (48.2,52.1] (28.3,32.3]
## [799] (28.3,32.3] (28.3,32.3] (32.3,36.2] (28.3,32.3] (8.38,12.4] (0.34,4.4]
## [805] (24.3,28.3] (28.3,32.3] (36.2,40.2] (16.3,20.3] (36.2,40.2] (32.3,36.2]
## [811] (24.3,28.3] (36.2,40.2] (32.3,36.2] (4.4,8.38]  (28.3,32.3] (24.3,28.3]
## [817] (20.3,24.3] (28.3,32.3] (40.2,44.2] (8.38,12.4] (48.2,52.1] (24.3,28.3]
## [823] (36.2,40.2] (24.3,28.3] (0.34,4.4]  (24.3,28.3] (24.3,28.3] (0.34,4.4]
## [829] (24.3,28.3] (60.1,64.1] (12.4,16.3] (0.34,4.4]  (24.3,28.3] (20.3,24.3]
## [835] (16.3,20.3] (36.2,40.2] (20.3,24.3] (24.3,28.3] (28.3,32.3] (24.3,28.3]
## [841] (16.3,20.3] (12.4,16.3] (28.3,32.3] (32.3,36.2] (16.3,20.3] (40.2,44.2]
## [847] (24.3,28.3] (32.3,36.2] (24.3,28.3] (24.3,28.3] (0.34,4.4]  (72,76]
## [853] (8.38,12.4] (12.4,16.3] (40.2,44.2] (16.3,20.3] (44.2,48.2] (48.2,52.1]
## [859] (20.3,24.3] (24.3,28.3] (40.2,44.2] (20.3,24.3] (44.2,48.2] (24.3,28.3]
## [865] (20.3,24.3] (40.2,44.2] (24.3,28.3] (28.3,32.3] (24.3,28.3] (0.34,4.4]
## [871] (24.3,28.3] (44.2,48.2] (32.3,36.2] (44.2,48.2] (24.3,28.3] (12.4,16.3]
## [877] (16.3,20.3] (16.3,20.3] (24.3,28.3] (52.1,56.1] (24.3,28.3] (32.3,36.2]
## [883] (20.3,24.3] (24.3,28.3] (24.3,28.3] (36.2,40.2] (24.3,28.3] (16.3,20.3]
## [889] (24.3,28.3] (24.3,28.3] (28.3,32.3]
## 20 Levels: (0.34,4.4] (4.4,8.38] (8.38,12.4] (12.4,16.3] ... (76,80.1]
```

```
#AgeGroup and Survived
sumAgeGroup <- summarize( group_by(ds, AgeGroup), n=length(AgeGroup), Survived=mean(Survived))
```

```
## Warning in mean.default(Survived): argument is not numeric or logical: returning
## NA

## Warning in mean.default(Survived): argument is not numeric or logical: returning
## NA

## Warning in mean.default(Survived): argument is not numeric or logical: returning
## NA

## Warning in mean.default(Survived): argument is not numeric or logical: returning
## NA

## Warning in mean.default(Survived): argument is not numeric or logical: returning
## NA

## Warning in mean.default(Survived): argument is not numeric or logical: returning
## NA

## Warning in mean.default(Survived): argument is not numeric or logical: returning
## NA

## Warning in mean.default(Survived): argument is not numeric or logical: returning
## NA

## Warning in mean.default(Survived): argument is not numeric or logical: returning
## NA

## Warning in mean.default(Survived): argument is not numeric or logical: returning
## NA

## Warning in mean.default(Survived): argument is not numeric or logical: returning
## NA

## Warning in mean.default(Survived): argument is not numeric or logical: returning
## NA

## Warning in mean.default(Survived): argument is not numeric or logical: returning
## NA

## Warning in mean.default(Survived): argument is not numeric or logical: returning
## NA

## Warning in mean.default(Survived): argument is not numeric or logical: returning
## NA

## Warning in mean.default(Survived): argument is not numeric or logical: returning
## NA

## Warning in mean.default(Survived): argument is not numeric or logical: returning
## NA

## Warning in mean.default(Survived): argument is not numeric or logical: returning
## NA

## Warning in mean.default(Survived): argument is not numeric or logical: returning
## NA

## Warning in mean.default(Survived): argument is not numeric or logical: returning
## NA

## `summarise()` ungrouping output (override with `.groups` argument)
```

```
gAgeGroup1 <- ds %>%
group_by(Survived, AgeGroup) %>%
tally() %>%
group_by(Survived) %>%
mutate(x = n / sum(n)) %>%
ggplot() +
geom_col(aes(
x = factor(AgeGroup),
y = x,
fill = factor(Survived)
), position = "stack")

grid.arrange(gAgeGroup1, nrow=1)
```


0.4
0.2
0.0
x
factor(AgeGroup)
factor(Survived)
🟥 Not
🟦 Yes