XAL2: Sácale jugo a tus datos

Sesión 2

Said Muñoz, Miguel Nuñez

6 de Agosto 2019

Introducción a Tidyverse

"The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures." https://www.tidyverse.org/

Instalación

Para instalar las bibliotecas, basta con ejecutar el siguiente comando:

```
install.packages(c("tidyverse","dplyr"))
```

Ejemplo 1

```
library(tidyverse)
library(dplyr)


df<-readr::read_csv("https://raw.githubusercontent.com/dataoptimal/posts/master/data%20cleaning%20with%2


df %>%
  filter(Churn=="yes")
```

```
## # A tibble: 5 x 5
##   customerID MonthlyCharges TotalCharges PaymentMethod    Churn
##   <chr>               <dbl> <chr>        <chr>            <chr>
## 1 7590-VHVEG           29.8 109.9        Electronic check yes
## 2 5575-GNVDE           57.0 na           Mailed check     yes
## 3 3668-QPYBK           NA   108.15       --               yes
## 4 9305-CDSKC          NaN   820.5        --               yes
## 5 6713-OKOMC           NA   N/A          <NA>             yes
```

```
# nested functions
log(sin(exp(2)))
```

```
## [1] -0.1122118
```

```
# piped functions
2 %>% exp() %>%
  sin() %>%
  log()
```

```
## [1] -0.1122118
```

```r
# filter on customers that churned,
# select customerID and TotalCharges columns
df %>%
  filter(Churn=="yes") %>%
  select(-c(customerID, TotalCharges))
```

```
## # A tibble: 5 x 3
##   MonthlyCharges PaymentMethod    Churn
##            <dbl> <chr>            <chr>
## 1           29.8 Electronic check yes
## 2           57.0 Mailed check     yes
## 3           NA   --               yes
## 4          NaN   --               yes
## 5           NA   <NA>             yes
```

```r
df$MonthlyCharges
```

```
##  [1]  29.85  56.95     NA  42.30  70.70    NaN  89.10     NA 104.80  54.10
```

```r
is.na(df$MonthlyCharges)
```

```
##  [1] FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE
```

```r
anyNA(df$MonthlyCharges)
```

```
## [1] TRUE
```

```r
df %>%
  distinct(MonthlyCharges)
```

```
## # A tibble: 9 x 1
##   MonthlyCharges
##            <dbl>
## 1           29.8
## 2           57.0
## 3           NA
## 4           42.3
## 5           70.7
## 6          NaN
## 7           89.1
## 8          105.
## 9           54.1
```

```r
# counting unique values
df %>%
  summarise(numero_de_elementos_unicos = n_distinct(MonthlyCharges),
            suma=sum(MonthlyCharges,na.rm = TRUE))
```

```
## # A tibble: 1 x 2
##   numero_de_elementos_unicos  suma
##                        <int> <dbl>
## 1                          9  448.
```

```r
# counting missing values
df %>%
  summarise(count = sum(is.na(MonthlyCharges)))
```

```
## # A tibble: 1 x 1
##   count
##   <int>
## 1     3
```

```
# counting unique, missing, and median values
df %>% summarise(n = n_distinct(MonthlyCharges),
                 na = sum(is.na(MonthlyCharges)),
                 med = median(MonthlyCharges, na.rm = TRUE))
```

```
## # A tibble: 1 x 3
##       n    na   med
##   <int> <int> <dbl>
## 1     9     3  57.0
```

```
# counting unique, missing, and median values
df %>% summarise(n = n_distinct(MonthlyCharges),
                 na = sum(is.na(MonthlyCharges)),
                 med = median(MonthlyCharges, na.rm = TRUE))
```

```
## # A tibble: 1 x 3
##       n    na   med
##   <int> <int> <dbl>
## 1     9     3  57.0
```

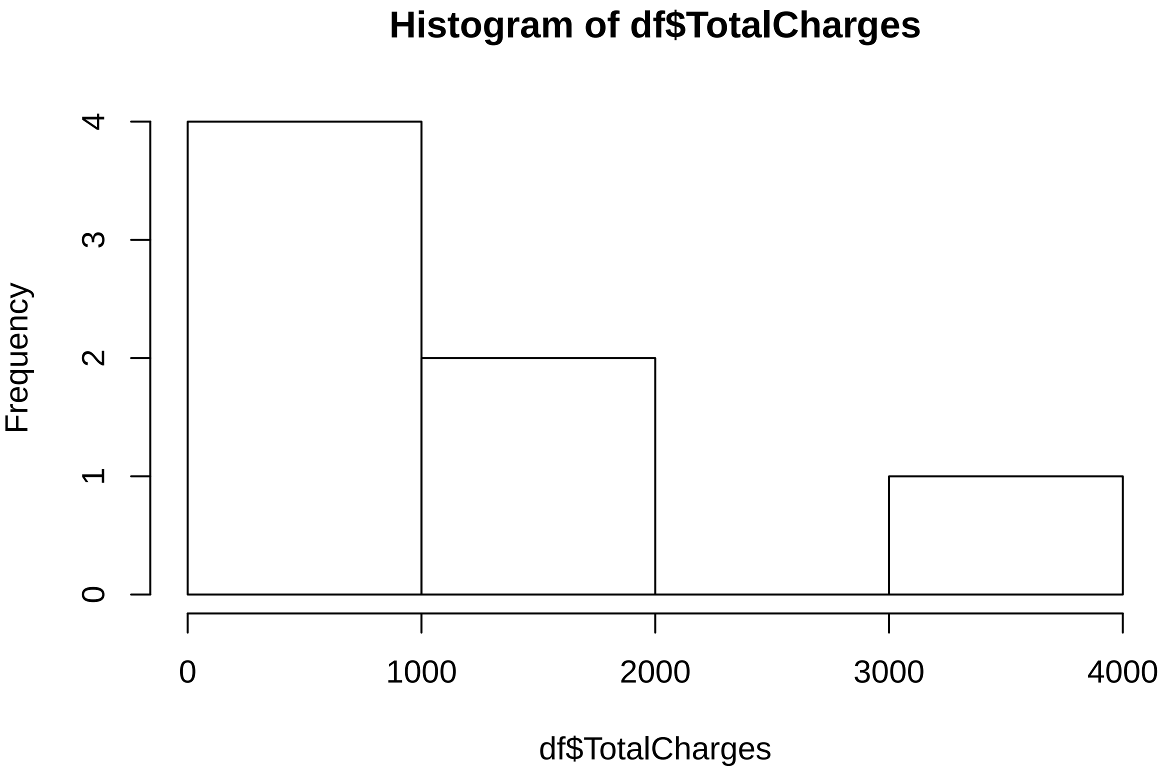
```
# mutate missing values
df<-df %>%
  mutate(CargosMensualesPesos
         = MonthlyCharges*19.5)

# mutate missing values
df %>%
  mutate(MonthlyCharges
         = replace(MonthlyCharges,
                   is.na(MonthlyCharges),
                   0
                   )
         )
```

```
## # A tibble: 10 x 6
##    customerID MonthlyCharges TotalCharges PaymentMethod Churn
##    <chr>               <dbl> <chr>        <chr>         <chr>
##  1 7590-VHVEG           29.8 109.9        Electronic c~ yes
##  2 5575-GNVDE           57.0 na           Mailed check  yes
##  3 3668-QPYBK            0   108.15       --            yes
##  4 7795-CFOCW           42.3 1840.75      Bank transfer no
##  5 9237-HQITU           70.7 <NA>         Electronic c~ no
##  6 9305-CDSKC            0   820.5        --            yes
##  7 1452-KIOVK           89.1 1949.4       Credit card   no
##  8 6713-OKOMC            0   N/A          <NA>          yes
##  9 7892-POOKP          105.  3046.05      Electronic c~ no
## 10 8451-AJOMK           54.1 354.95       Electronic c~ no
## # ... with 1 more variable: CargosMensualesPesos <dbl>
```

```
palabrasEliminar<-c("na","N/A")

df<-df %>%
  mutate(TotalChargesModificada = replace(TotalCharges, TotalCharges == palabrasEliminar , NA)) %>%
  mutate(TotalCharges = replace(TotalCharges, TotalCharges == "N/A", NA))

# taking another look
#df$TotalCharges
#is.na(df$TotalCharges)

df$TotalCharges <- as.numeric(df$TotalCharges)

glimpse(df$TotalCharges)
```

```
##  num [1:10] 110 NA 108 1841 NA ...
```

```
hist(df$TotalCharges)
```



```
# replace missing values with median
df <- df %>%
  mutate(TotalCharges = replace(TotalCharges,
                                is.na(TotalCharges),
                                median(TotalCharges, na.rm = T)))
df$TotalCharges
```

```
##  [1]  109.90  820.50  108.15 1840.75  820.50  820.50 1949.40  820.50
##  [9] 3046.05  354.95
```

```
## Otros problemas además de NA

# looking at PaymentMethod
df$PaymentMethod
```

```
##  [1] "Electronic check" "Mailed check"     "--"
##  [4] "Bank transfer"    "Electronic check" "--"
##  [7] "Credit card"      NA                 "Electronic check"
## [10] "Electronic check"
```

```
is.na(df$PaymentMethod)
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
```

```
# replacing "--" with NA
df <- df %>%
  mutate(PaymentMethod = replace(PaymentMethod, PaymentMethod ==  "--", NA))

is.na(df$PaymentMethod)
```

```
##  [1] FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE
```

```
df$PaymentMethod
```

```
##  [1] "Electronic check" "Mailed check"     NA                
##  [4] "Bank transfer"    "Electronic check" NA                
##  [7] "Credit card"      NA                 "Electronic check"
## [10] "Electronic check"
```

```
df$PaymentMethod
```

```
##  [1] "Electronic check" "Mailed check"     NA                
##  [4] "Bank transfer"    "Electronic check" NA                
##  [7] "Credit card"      NA                 "Electronic check"
## [10] "Electronic check"
```

```
table(df$PaymentMethod)
```

```
## 
##    Bank transfer      Credit card Electronic check     Mailed check 
##                1                1                4                1
```

```
df %>%
  mutate(PaymentMethod = replace(PaymentMethod, is.na(PaymentMethod), "unavailable"))
```

```
## # A tibble: 10 x 7
##    customerID MonthlyCharges TotalCharges PaymentMethod Churn
##    <chr>               <dbl>        <dbl> <chr>         <chr>
##  1 7590-VHVEG           29.8         110. Electronic c~ yes  
##  2 5575-GNVDE           57.0         820. Mailed check  yes  
##  3 3668-QPYBK           NA           108. unavailable   yes  
##  4 7795-CFOCW           42.3        1841. Bank transfer no   
##  5 9237-HQITU           70.7         820. Electronic c~ no   
##  6 9305-CDSKC          NaN           820. unavailable   yes  
##  7 1452-KIOVK           89.1        1949. Credit card   no   
##  8 6713-OKOMC           NA           820. unavailable   yes  
##  9 7892-POOKP          105.         3046. Electronic c~ no   
## 10 8451-AJOMK           54.1         355. Electronic c~ no   
## # ... with 2 more variables: CargosMensualesPesos <dbl>,
## #   TotalChargesModificada <chr>
```

Titanic

Introducción a ggplot2

gg se debe a `Grammar of Graphics`

```
library("ggplot2")
```

Dataset

```
data(population, package = "tidyr")
head(population)
```

```
## # A tibble: 6 x 3
##   country      year population
##   <chr>       <int>      <int>
## 1 Afghanistan  1995   17586073
## 2 Afghanistan  1996   18415307
## 3 Afghanistan  1997   19021226
## 4 Afghanistan  1998   19496836
## 5 Afghanistan  1999   19987071
## 6 Afghanistan  2000   20595360
```

```
tidy1<-head(population,100)
```

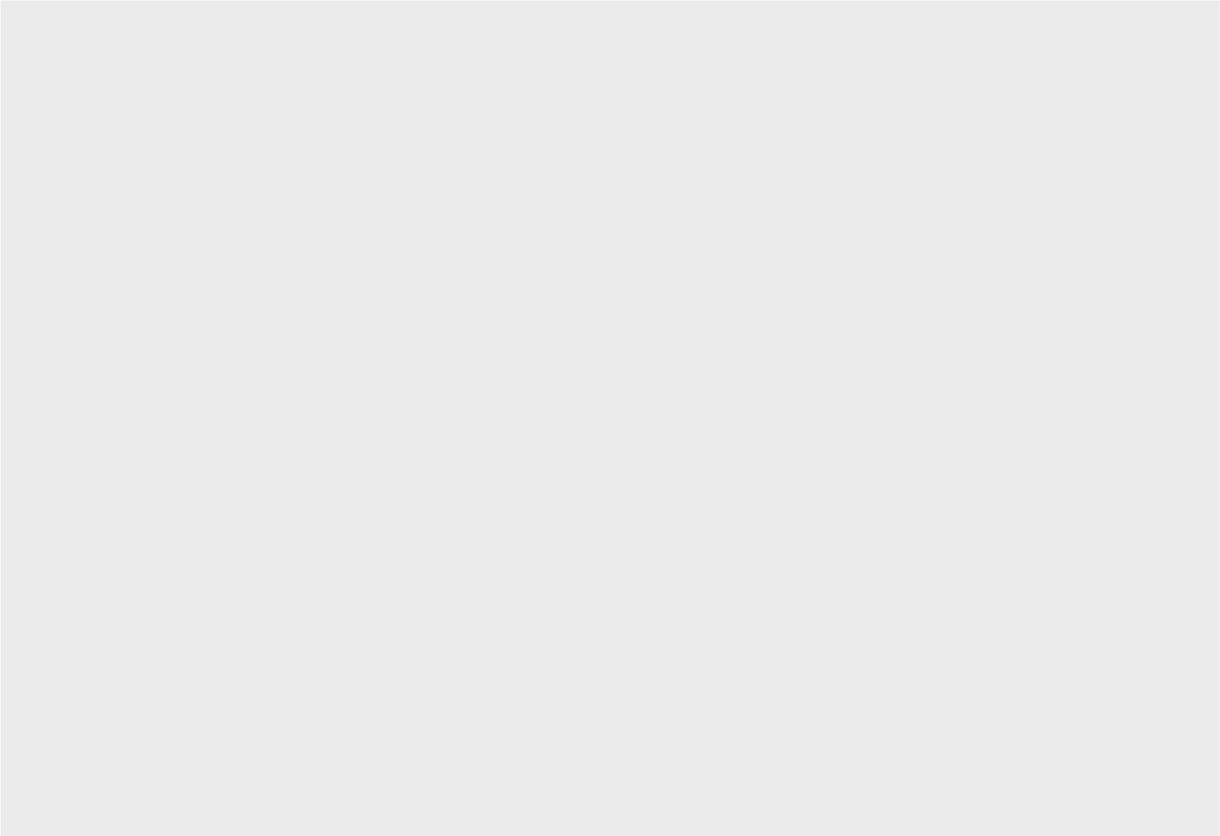
Algunas geom conocidas: - geom_point() - geom_line() - geom_bar() - geom_histogram() - geom_smooth() - geom_boxplot() - geom_text() - geom_{vh}line() - geom_count() - geom_density()

[1]

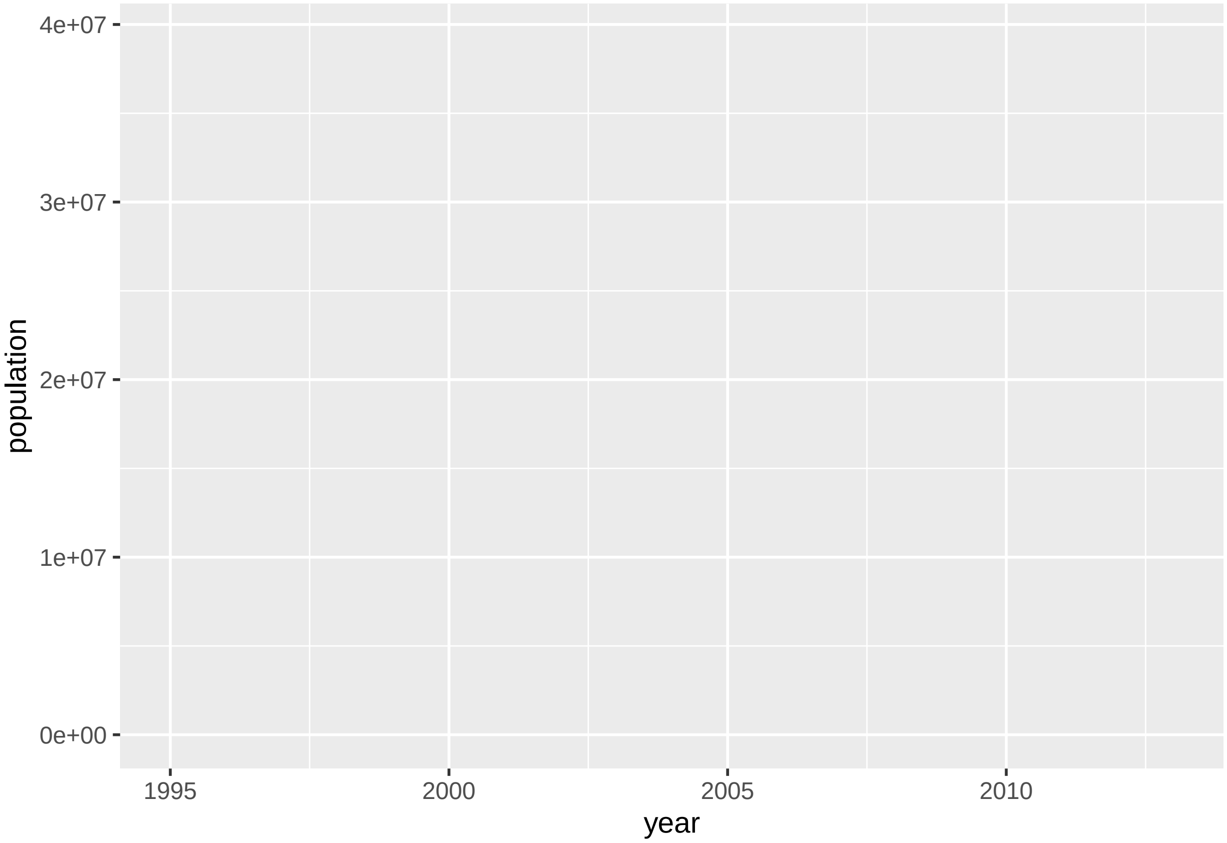
Plots

```
ggplot(tidy1)
```

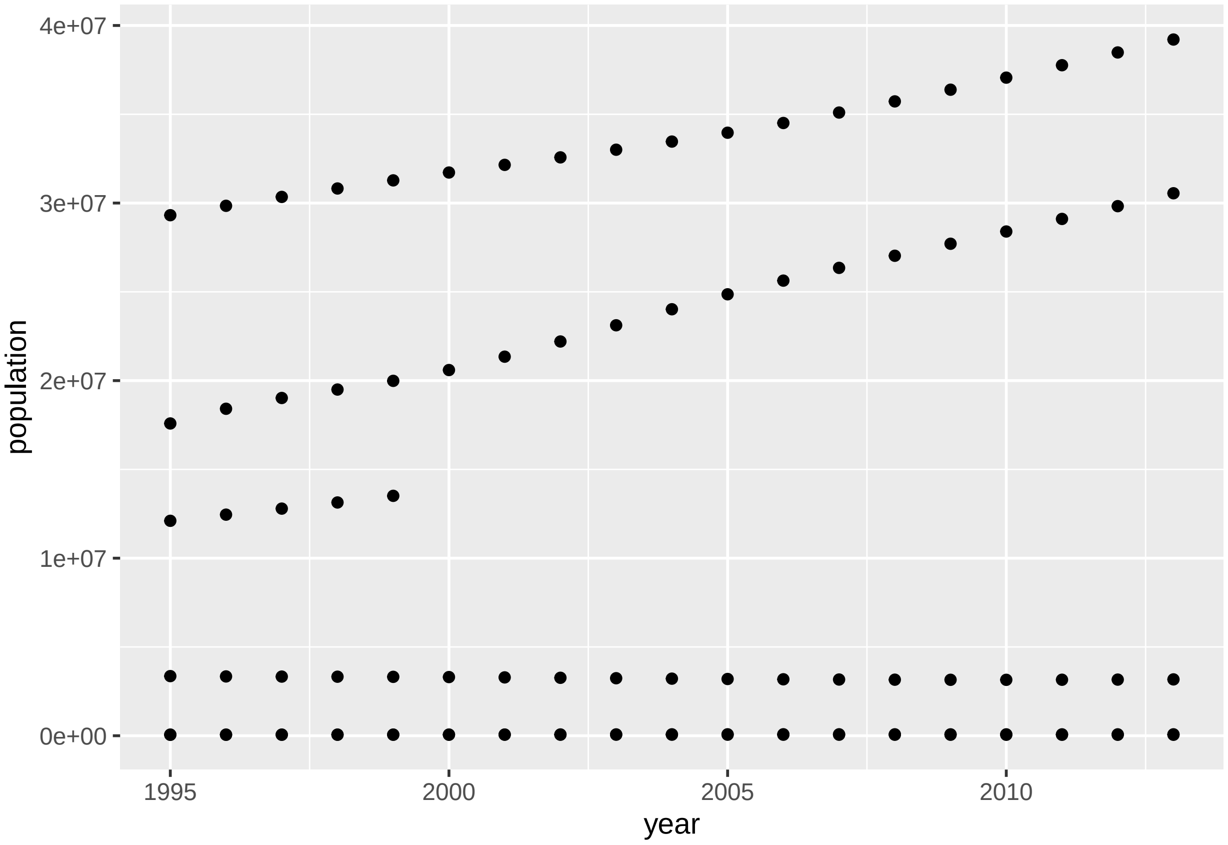
[1] https://eric.netlify.com/2017/08/10/most-popular-ggplot2-geoms/



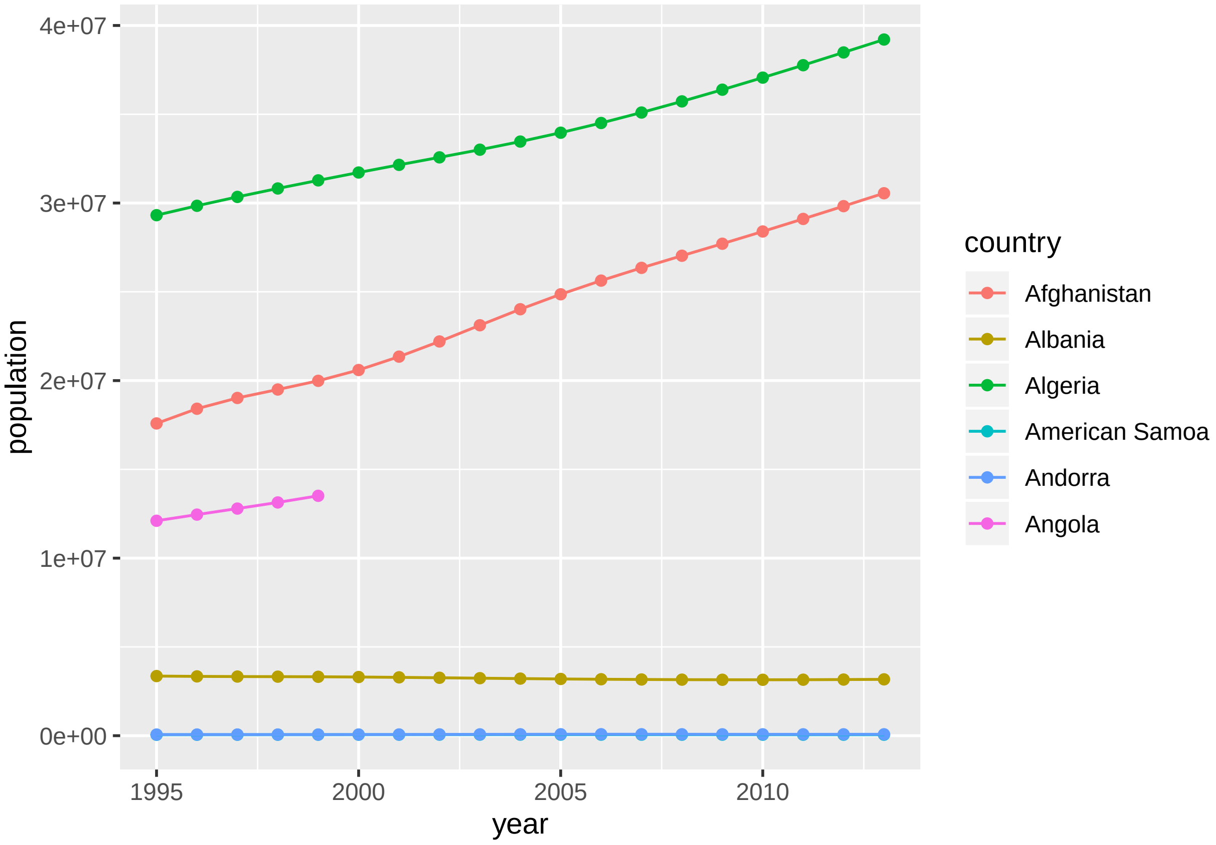
```r
ggplot(tidy1) + aes(x=year,
                    y=population)
```



```
ggplot(tidy1) + aes(x=year,
                    y=population) +
  geom_point()
```



```
ggplot(tidy1) + aes(x=year,
                    y=population,
                    color=country) +
  geom_point() +
  geom_line()
```

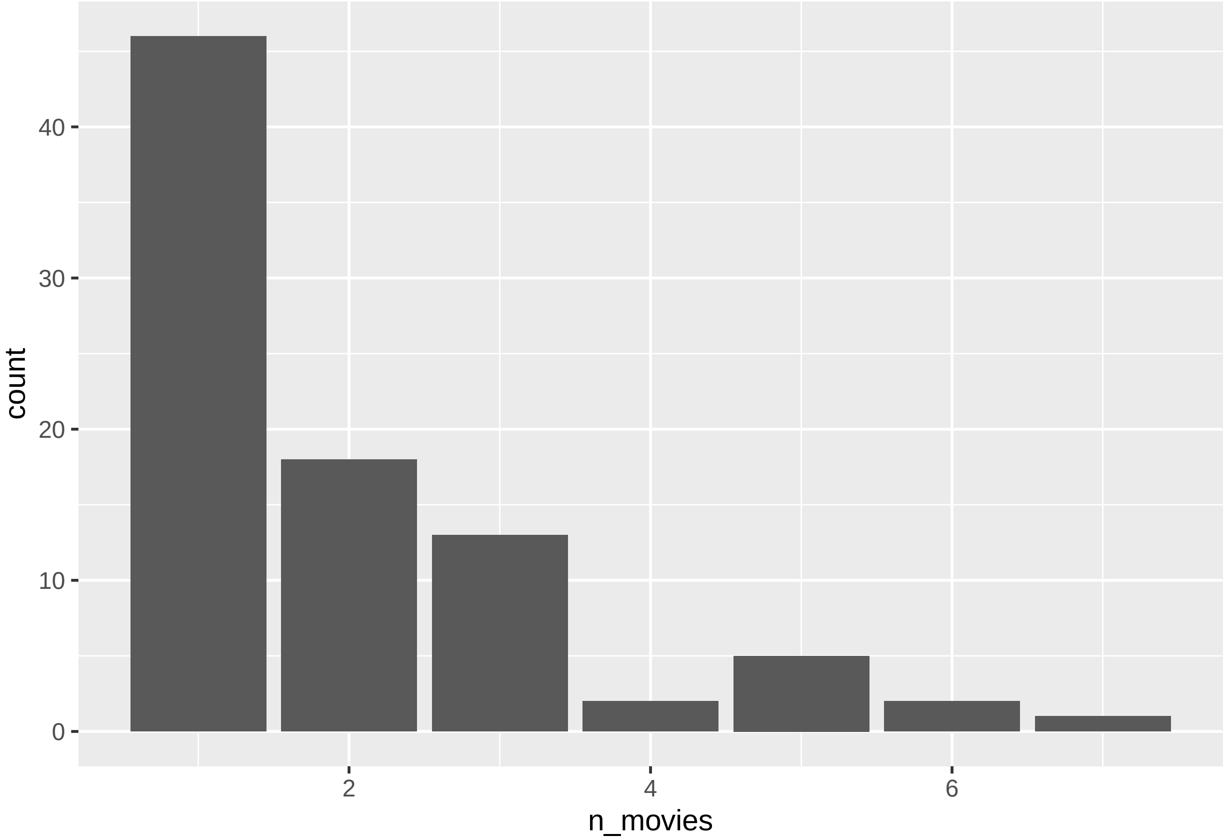


Ejemplo Star wars

```
library(tidyverse)
library(dplyr)
sw_chars <- starwars %>%
  mutate(
    n_movies = map_int(films, length),
    gender = ifelse(
      !gender %in% c('female', 'male'),
      'other', gender)
  ) %>%
  select(name, gender, n_movies)

sw_chars
```

```
## # A tibble: 87 x 3
##    name               gender n_movies
##    <chr>              <chr>     <int>
##  1 Luke Skywalker     male          5
##  2 C-3PO              other         6
##  3 R2-D2              other         7
##  4 Darth Vader        male          4
##  5 Leia Organa        female        5
##  6 Owen Lars          male          3
##  7 Beru Whitesun lars female        3
##  8 R5-D4              other         1
##  9 Biggs Darklighter  male          1
## 10 Obi-Wan Kenobi     male          6
## # ... with 77 more rows
```

```
ggplot(sw_chars) +
  aes(x = n_movies) +
  geom_bar(stat = "count")
```



```
install.packages("plotly")
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/3.6'
## (as 'lib' is unspecified)
```

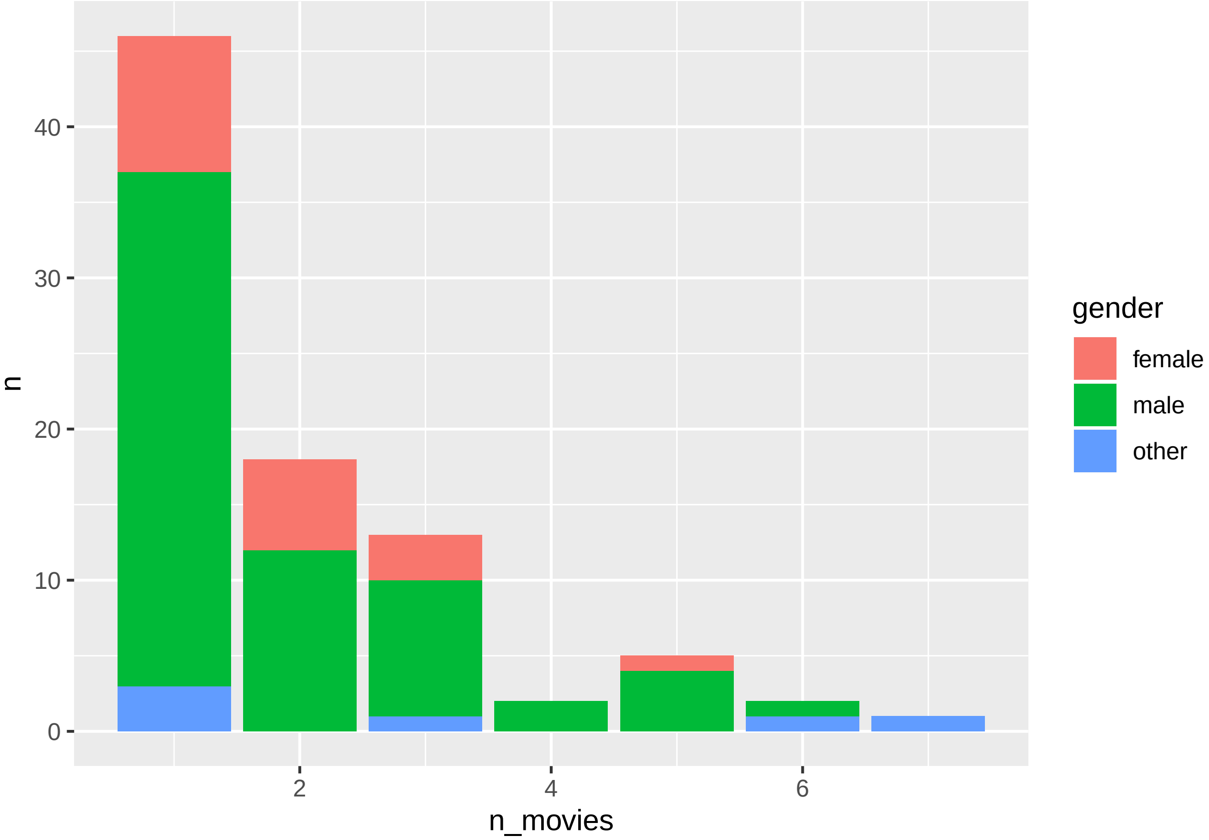
```
library("plotly")
```

```
##
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':
##
##     last_plot

## The following object is masked from 'package:stats':
##
##     filter

## The following object is masked from 'package:graphics':
##
##     layout
```

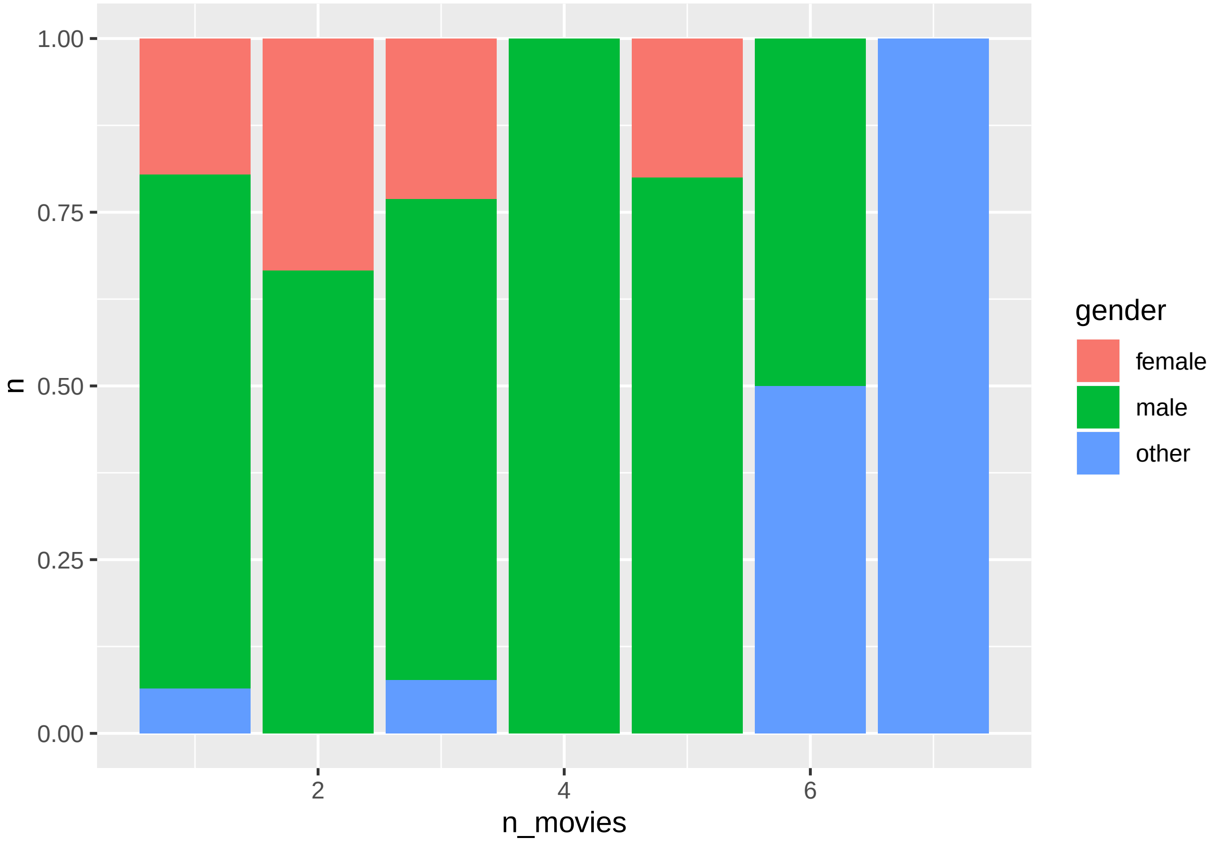
```r
plotPeliculas<-ggplot(sw_chars) +
  aes(x = n_movies,
      fill = gender) +
  geom_bar(stat = "count")
ggplotly(plotPeliculas)
```

```r
sw_chars_id <- sw_chars %>%
  group_by(n_movies, gender) %>%
  tally
```

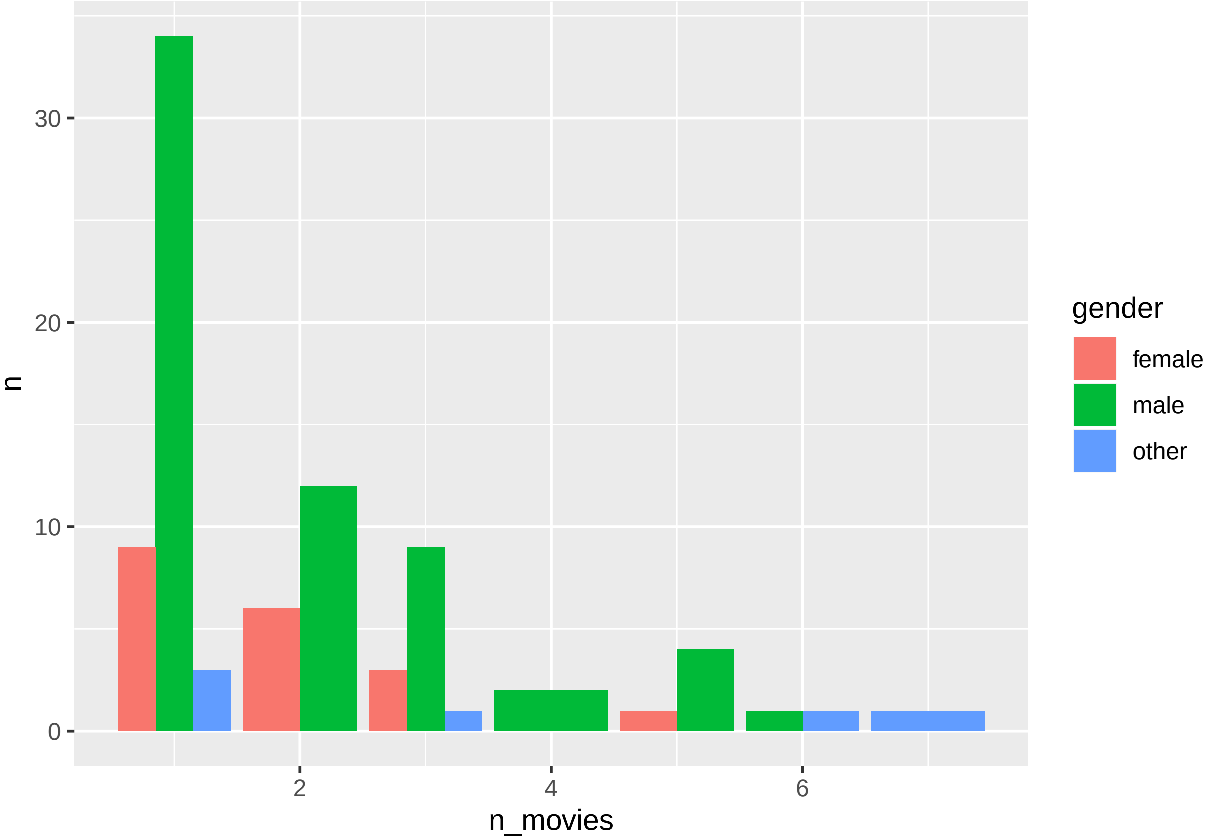
```r
ggplot(sw_chars_id) +
  aes(x = n_movies,
      y = n,
      fill = gender) +
  geom_bar(stat = 'identity')
```



```r
ggplot(sw_chars_id) +
  aes(x = n_movies,
      y = n,
      fill = gender) +
  geom_col(position = "fill")
```

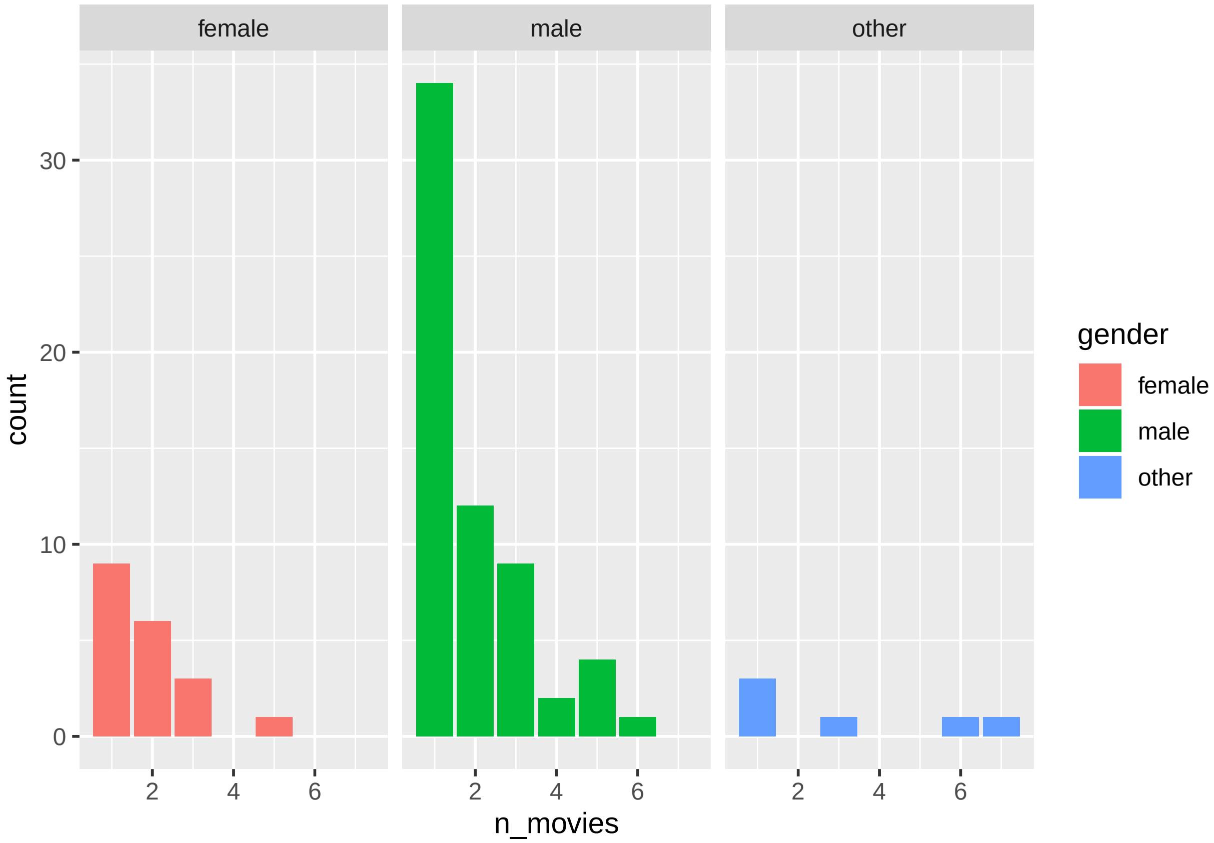


```r
ggplot(sw_chars_id) +
  aes(x = n_movies,
      y = n,
      fill = gender) +
  geom_col(position = "dodge")
```



```
g <- ggplot(sw_chars) +
  aes(x = n_movies,
      fill = gender) +
  geom_bar()
```

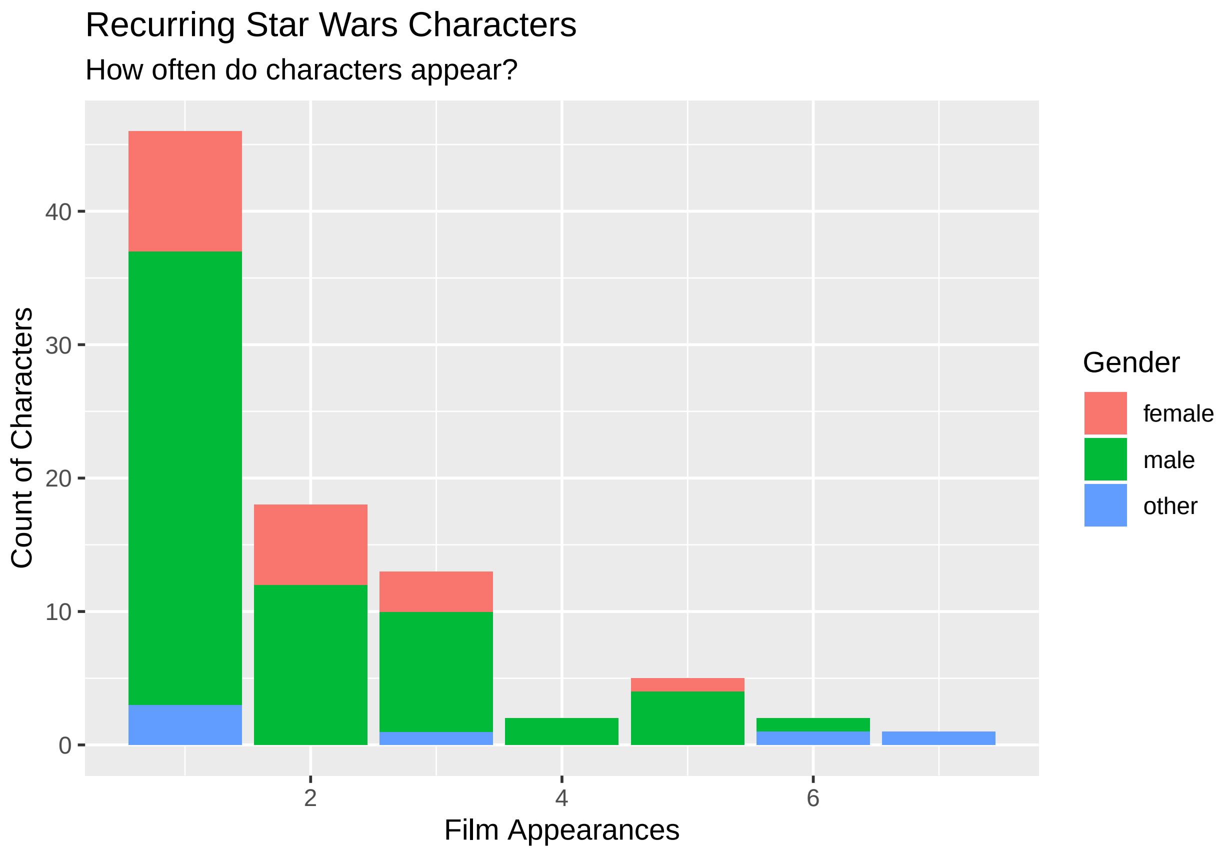
```
g + facet_wrap(~ gender)
```



```
g + facet_grid(gender ~ n_movies)
```

```
g + facet_grid(gender ~ n_movies, scales = 'free_y')
```

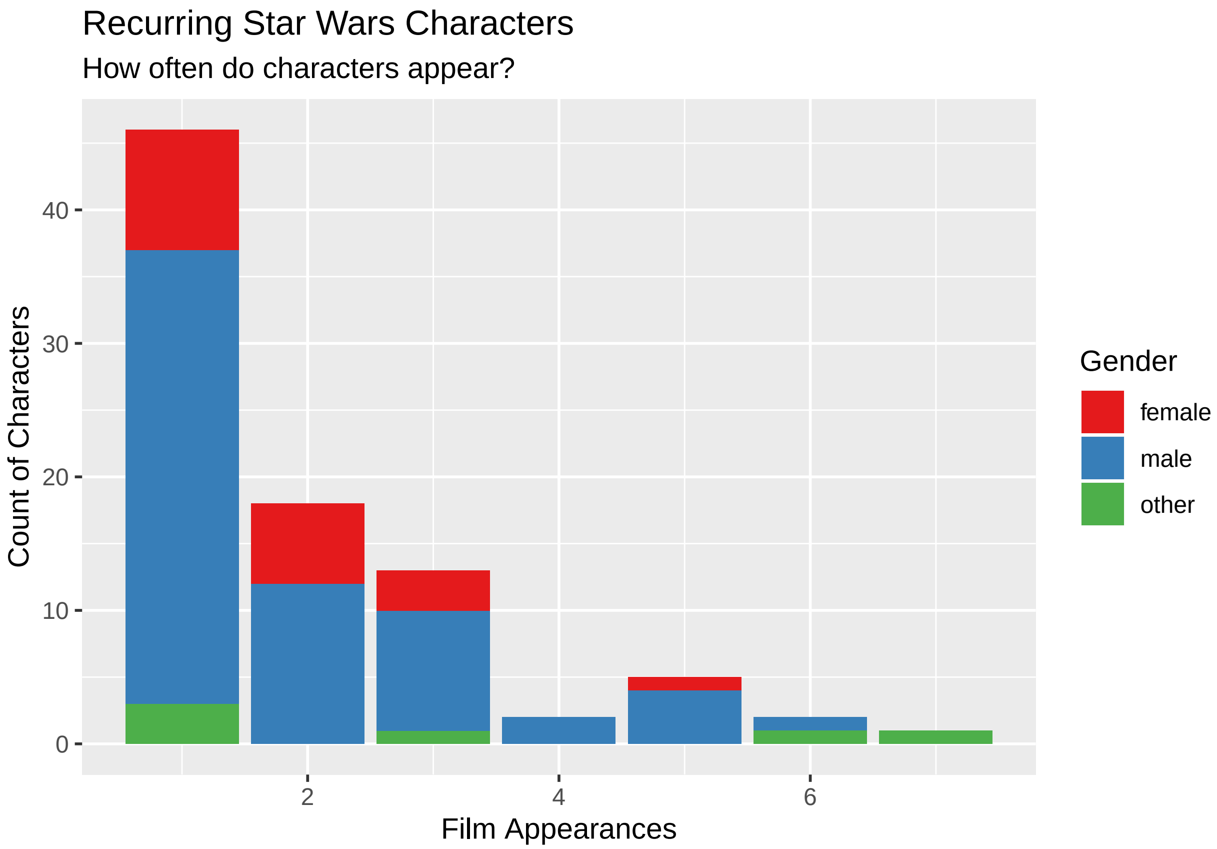
```
g <- g +
  labs(
    x = "Film Appearances",
    y = "Count of Characters",
    title = "Recurring Star Wars Characters",
    subtitle = "How often do characters appear?",
    fill = "Gender"
  )

g
```



Escalas

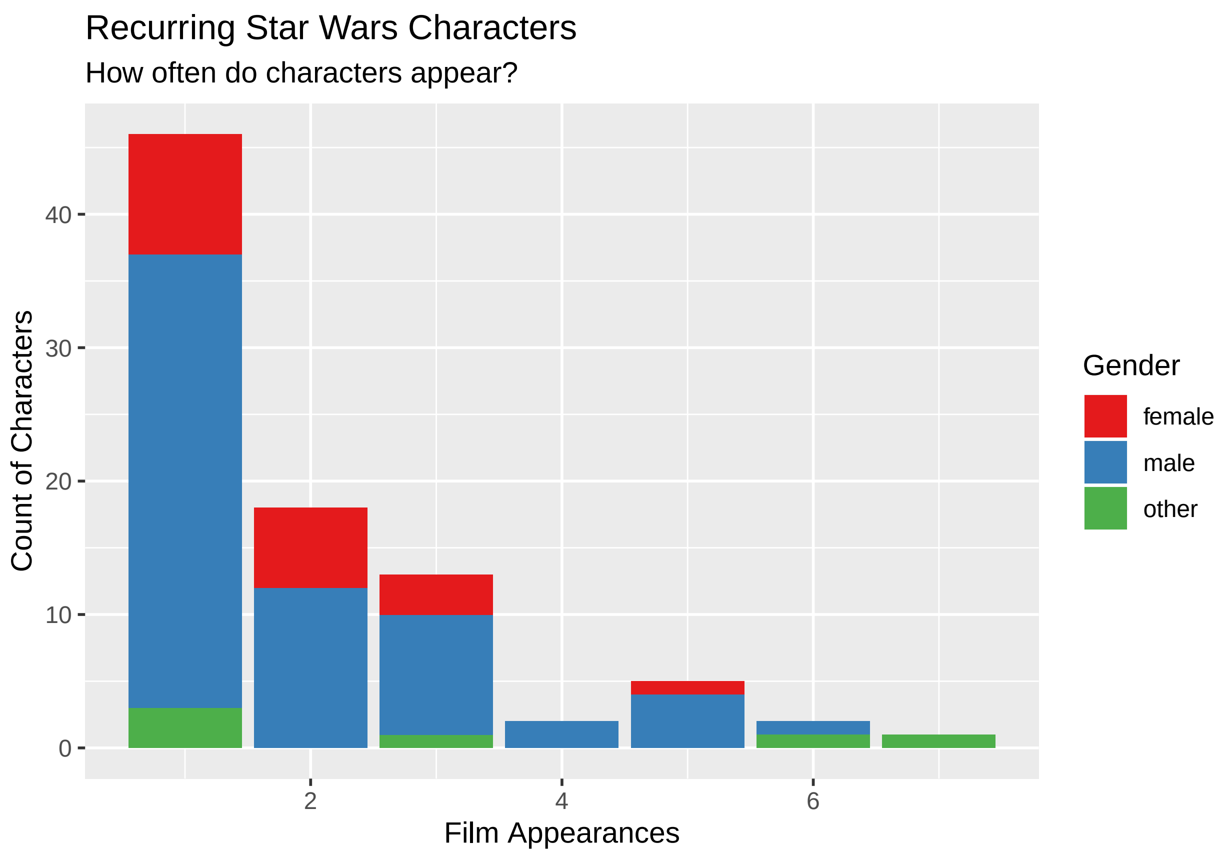
scale + _ + + _ + + ()

```
g <- g + scale_fill_brewer(palette = 'Set1')

g
```

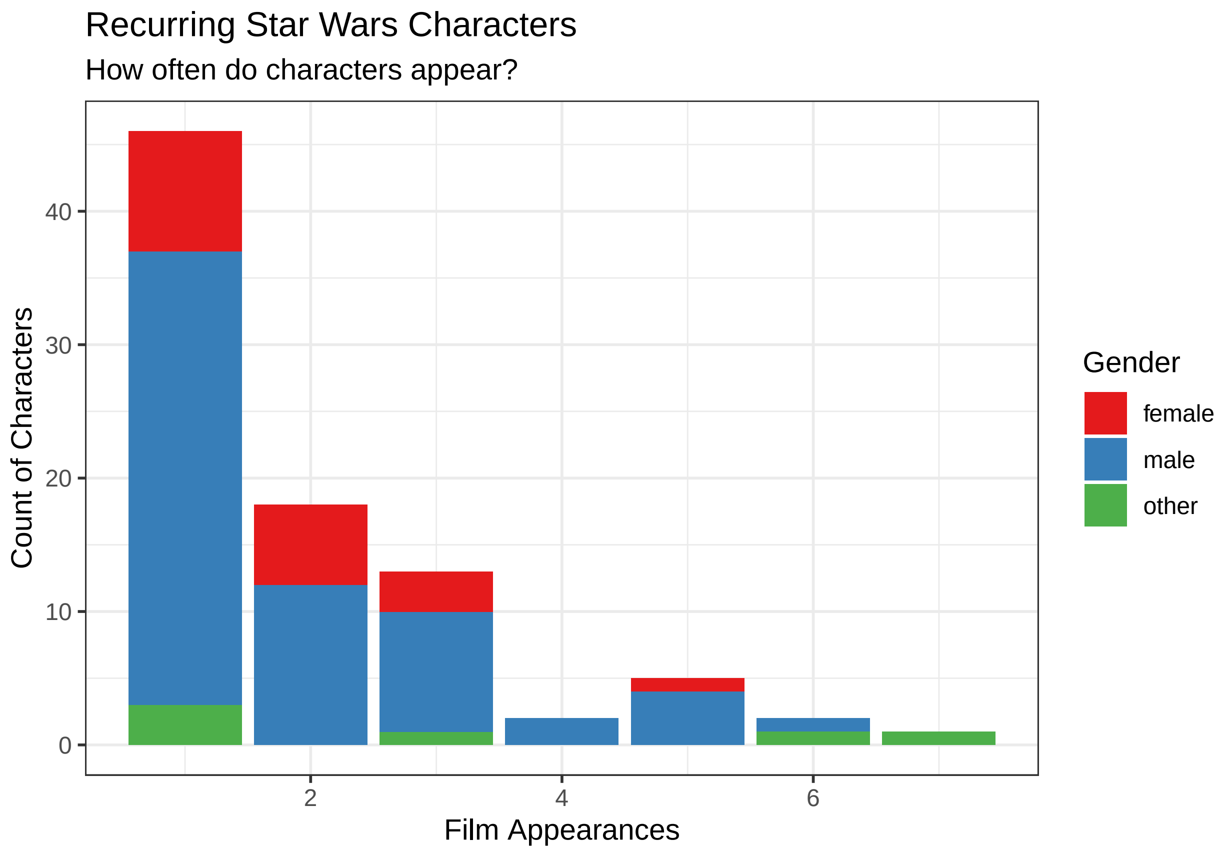

Recurring Star Wars Characters
How often do characters appear?
Count of Characters
0
10
20
30
40
Film Appearances
2
4
6
Gender
female
male
other

Temas

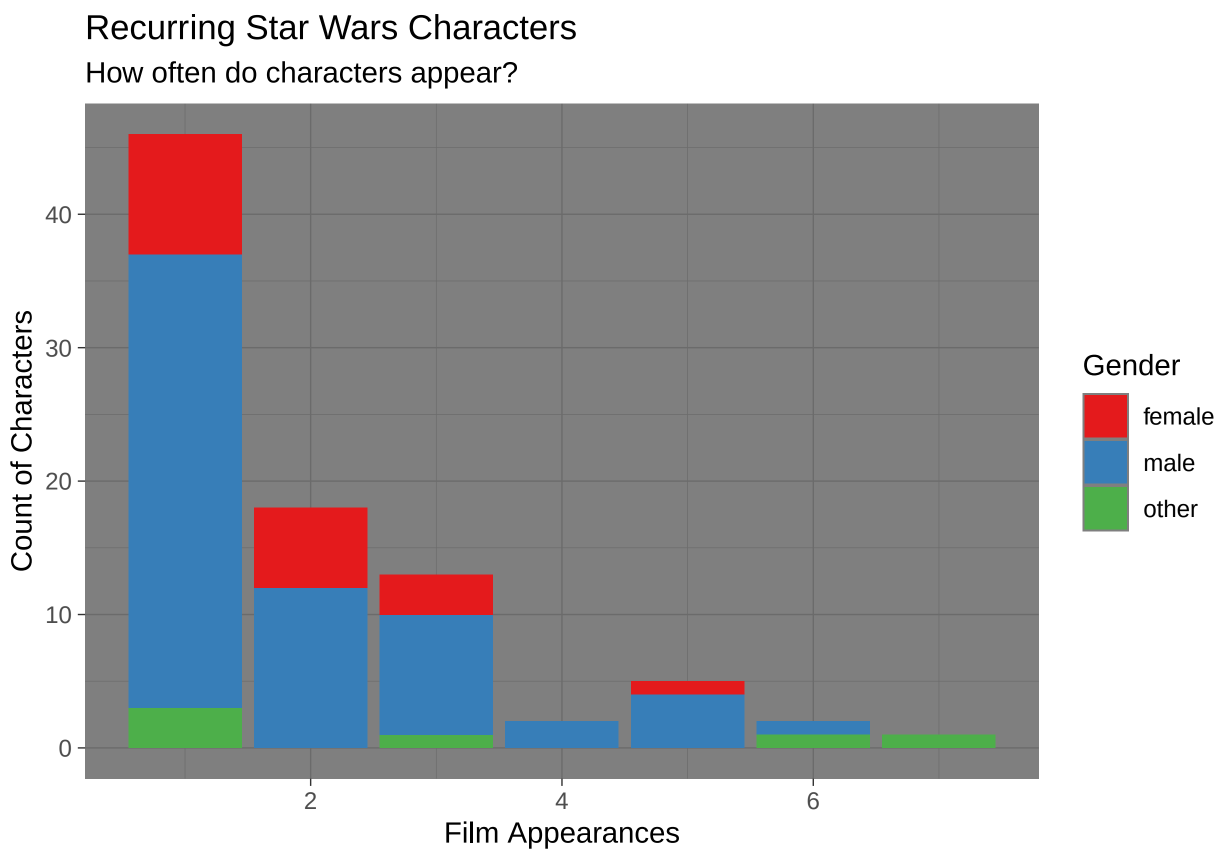
```
g
```


Recurring Star Wars Characters
How often do characters appear?
Count of Characters
0
10
20
30
40
Film Appearances
2
4
6
Gender
female
male
other

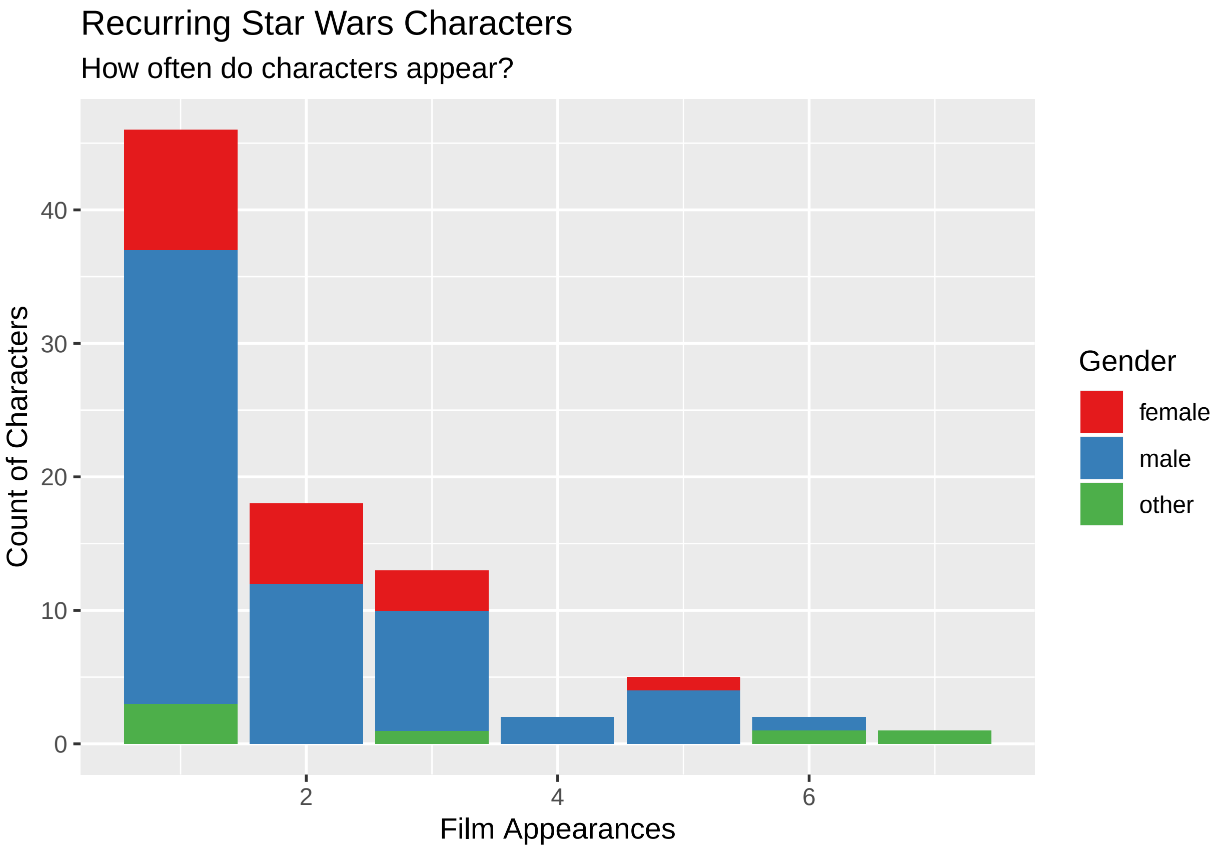
```
g + theme_bw()
```


Recurring Star Wars Characters
How often do characters appear?
Count of Characters
40
30
20
10
0
2
4
6
Film Appearances
Gender
female
male
other

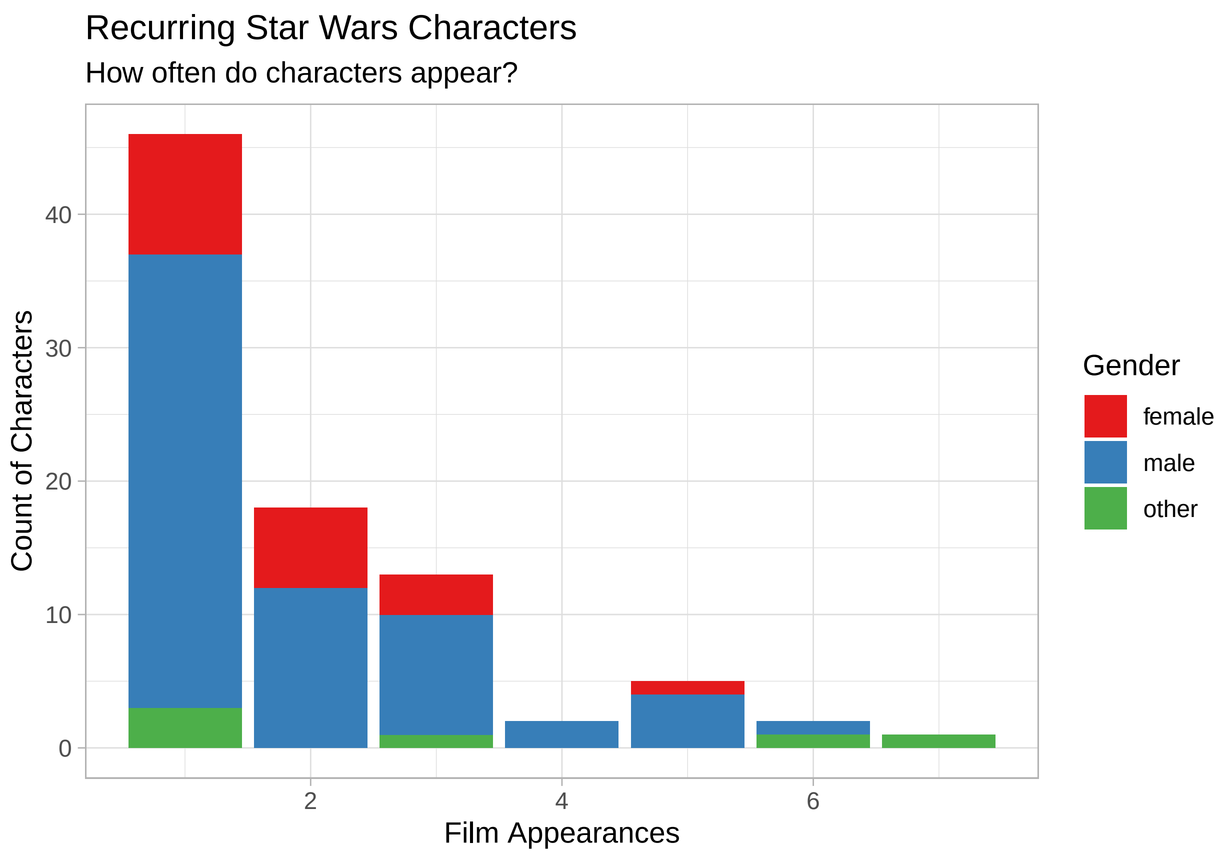
```
g + theme_dark()
```



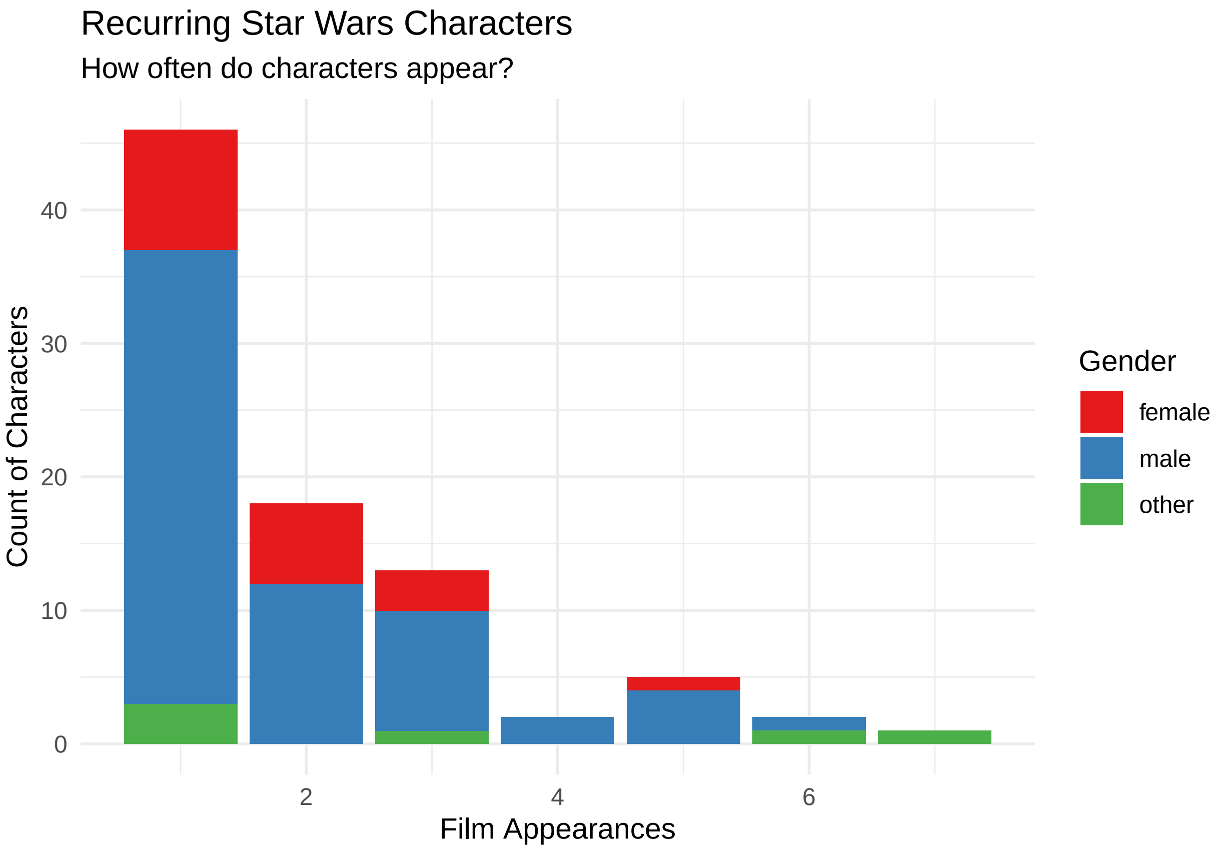
```r
g + theme_gray()
```



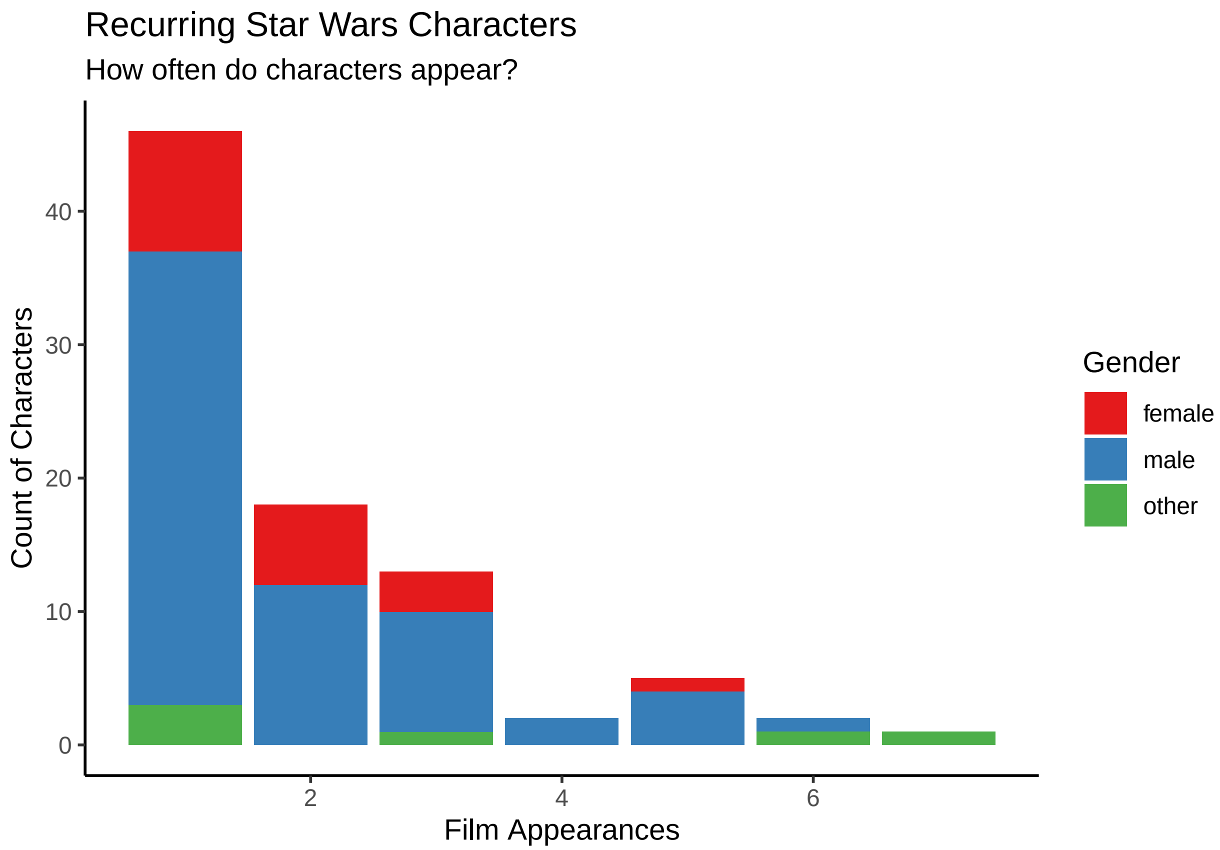
```r
g + theme_light()
```



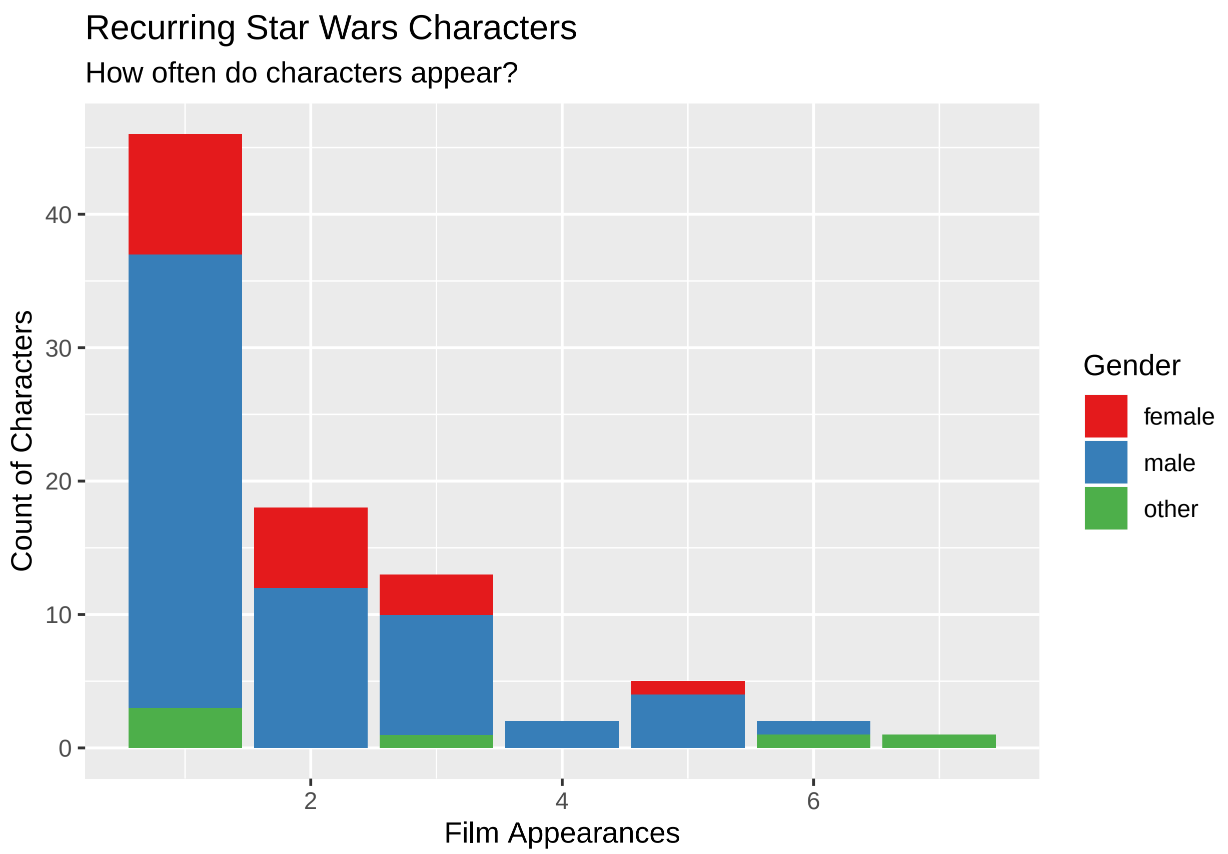
```
g + theme_minimal()
```



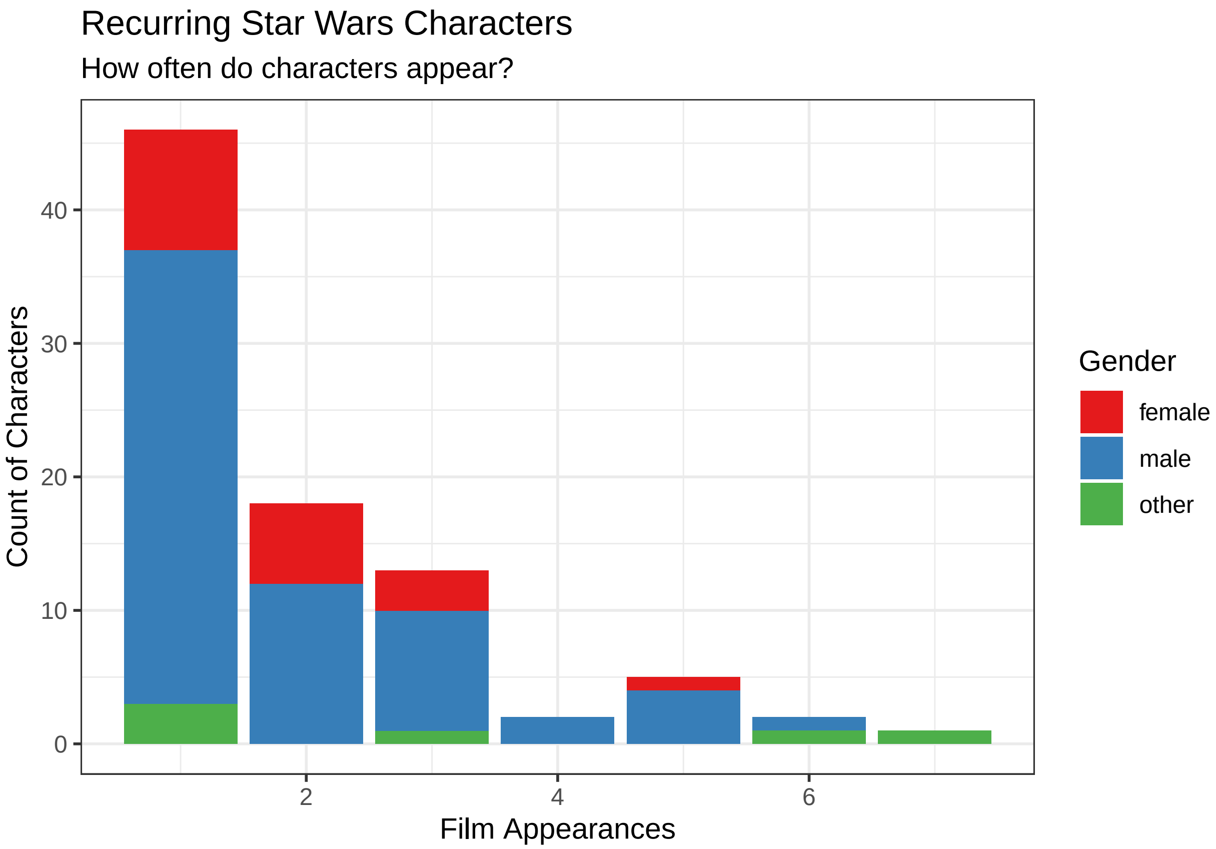
```
g+theme_classic()
```


Recurring Star Wars Characters
How often do characters appear?
Count of Characters
0
10
20
30
40
2
4
6
Film Appearances
Gender
🟥 female
🟦 male
🟩 other

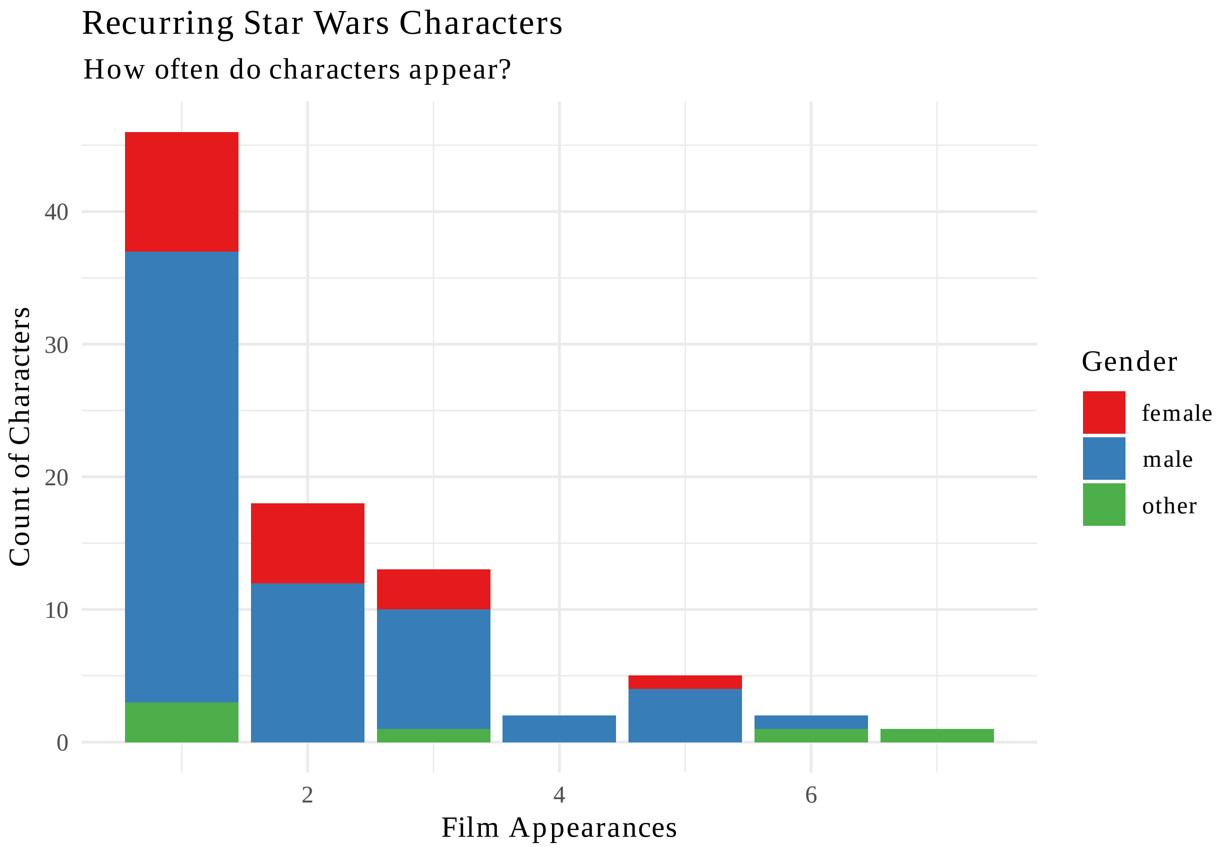
```
g
```


Recurring Star Wars Characters
How often do characters appear?
Count of Characters
0
10
20
30
40
2
4
6
Film Appearances
Gender
female
male
other

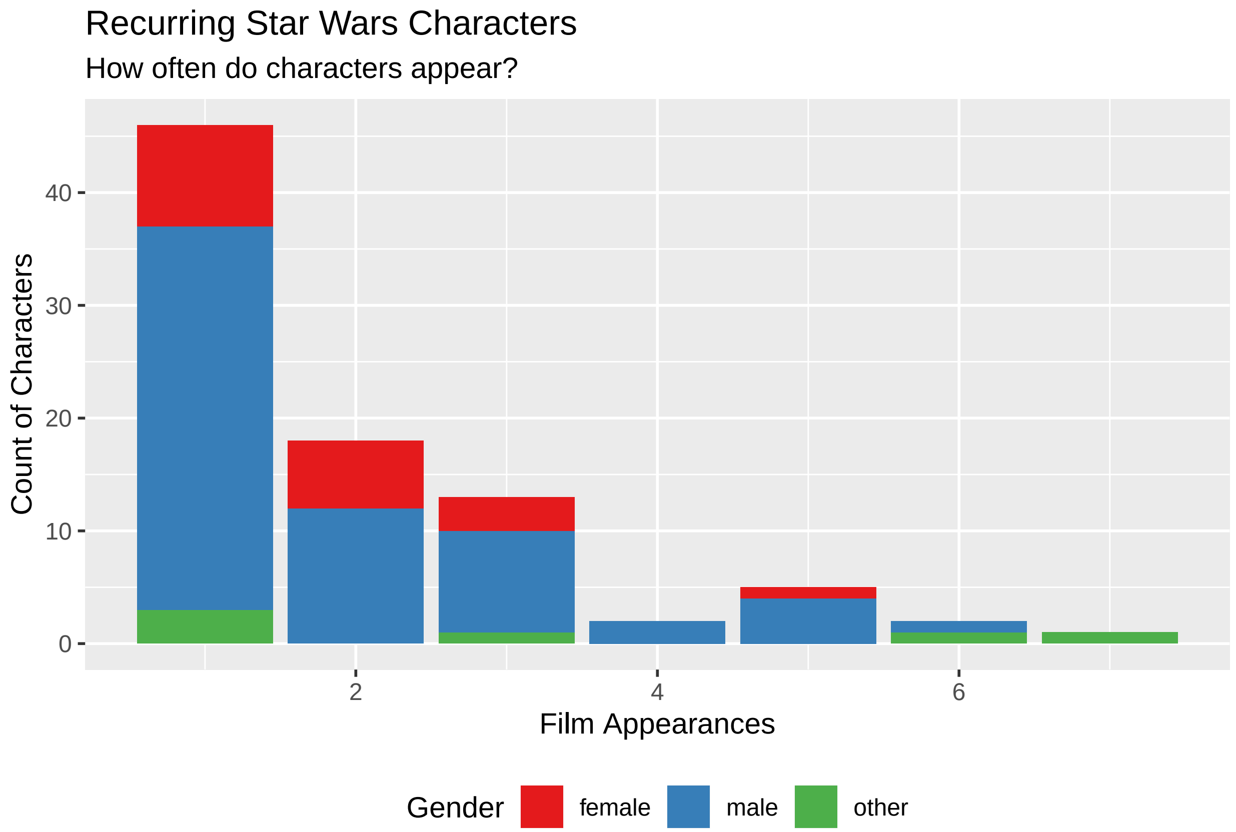
```
g + theme_bw()
```



```r
g + theme_minimal() + theme(text = element_text(family = "Palatino"))
```



```r
g + theme(legend.position = 'bottom')
```



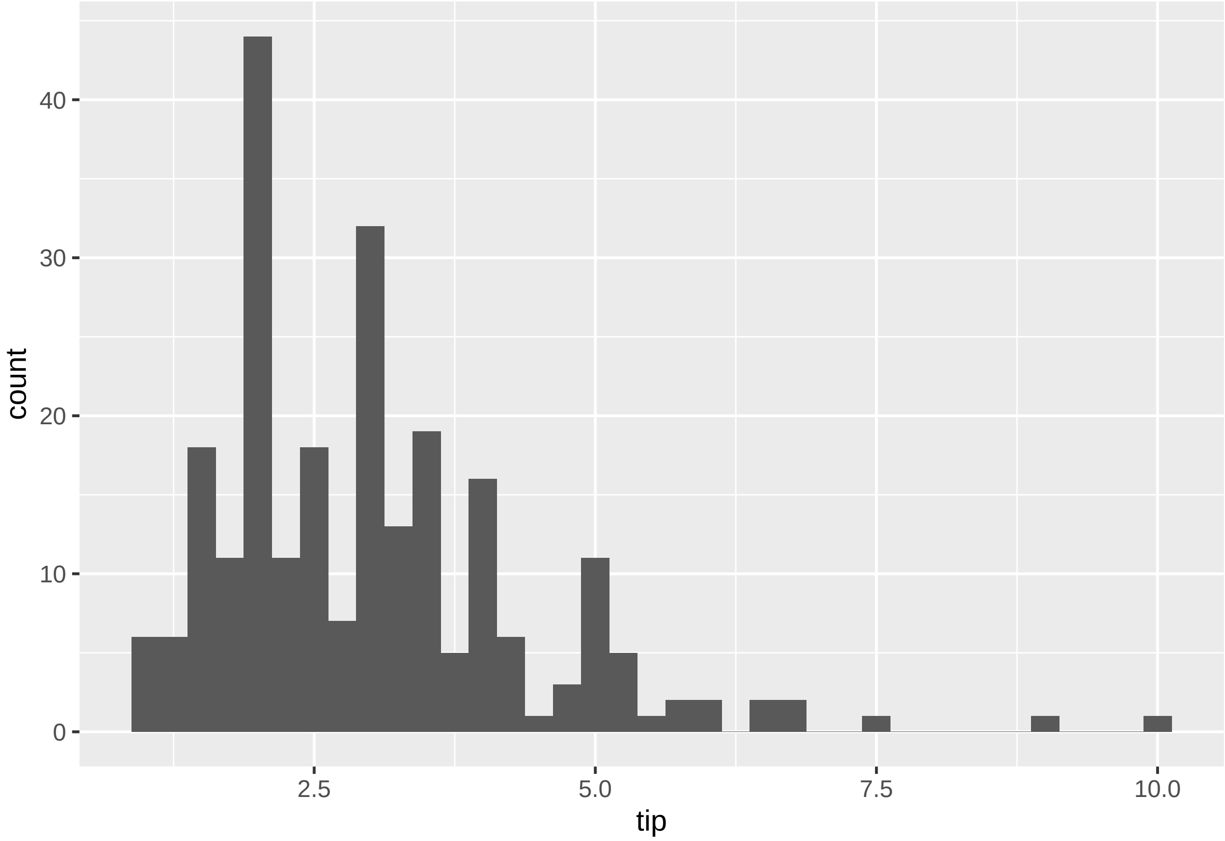
```r
g <- g +
  theme_minimal(base_family = 'Palatino') +
  theme(
    axis.text.y = element_blank(),
    strip.text = element_text(size = 18, face = 'bold'),
    panel.grid.major.y = element_blank(),
    panel.grid.minor.y = element_blank(),
    panel.grid.minor.x = element_blank(),
    panel.grid.major.x = element_line(color = "grey80", linetype = 3))
```

Tips

```r
library("reshape2")
```

```
##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths
```

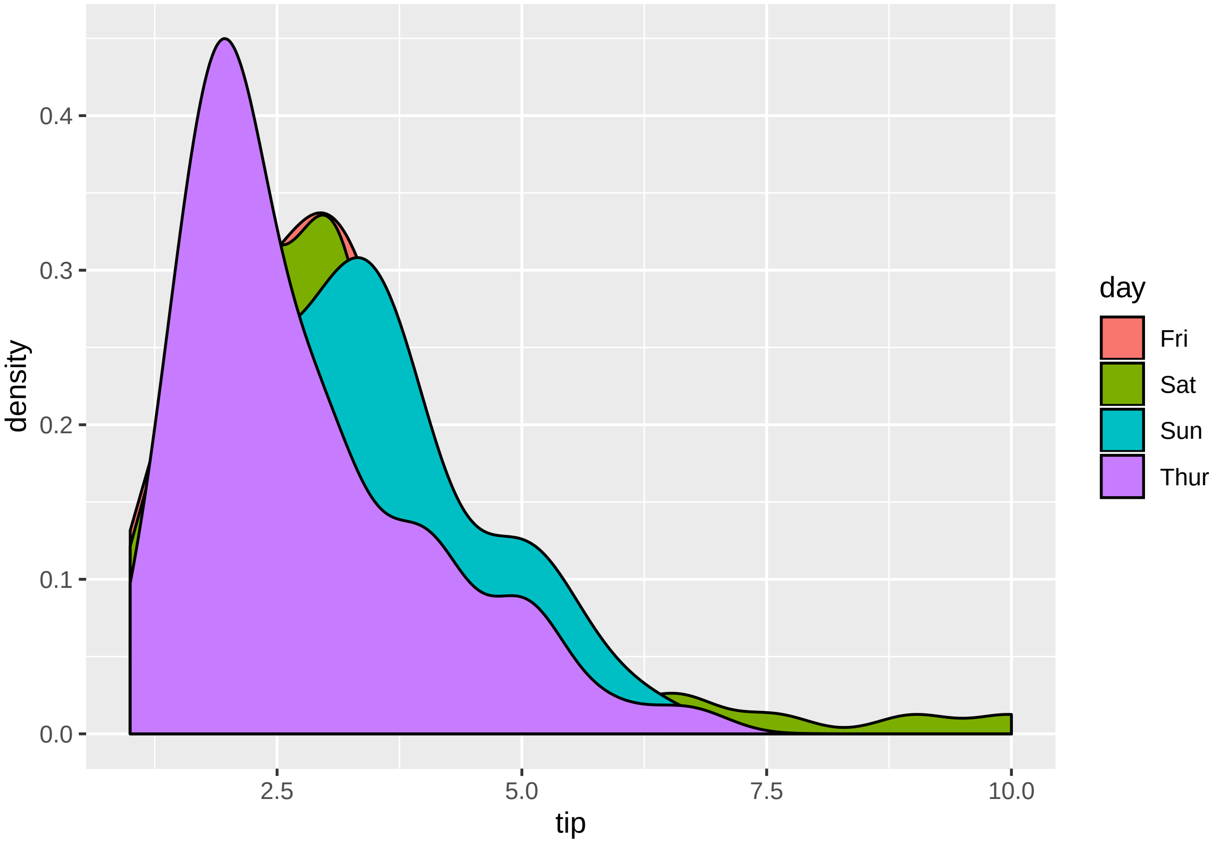
```r
ggplot(tips) +
  aes(x = tip) +
  geom_histogram(
    binwidth = 0.25
  )
```



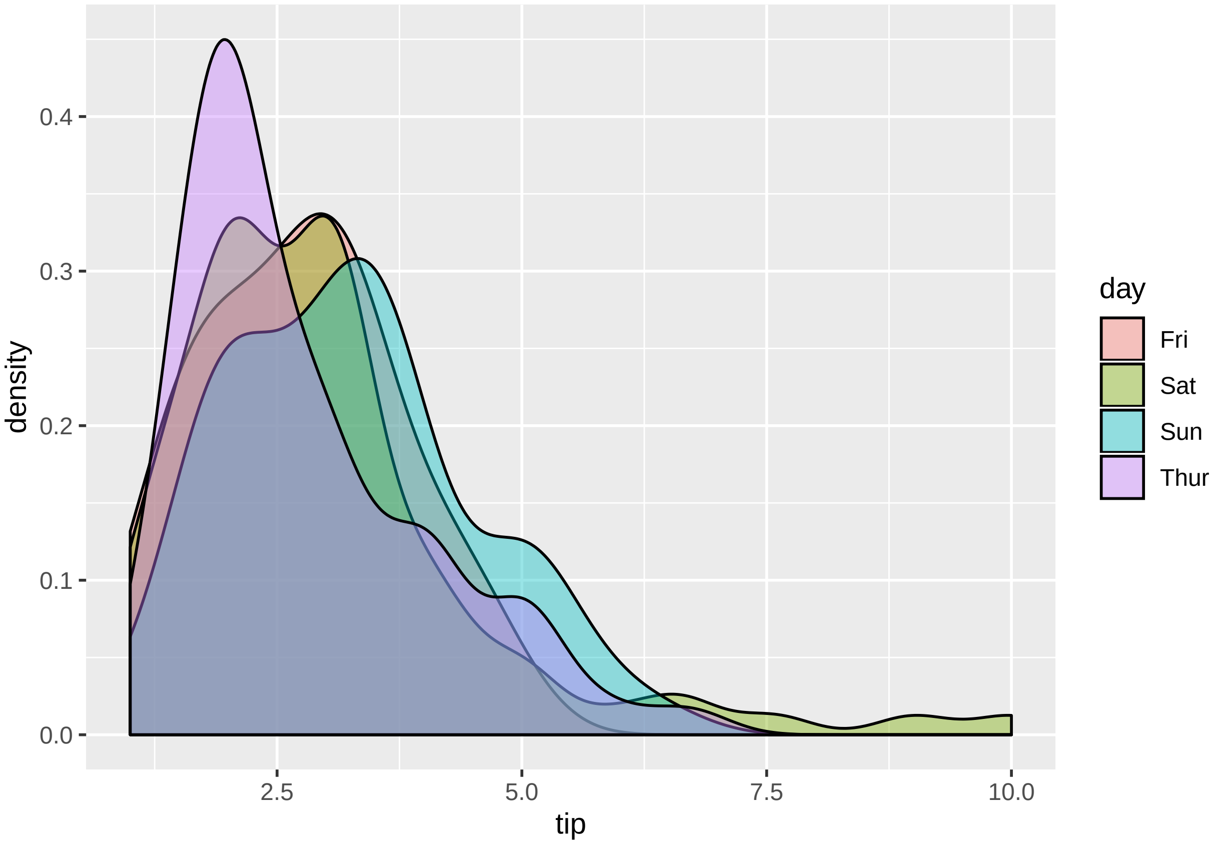
```
tips$tip
```

```
##   [1]  1.01  1.66  3.50  3.31  3.61  4.71  2.00  3.12  1.96  3.23  1.71
##  [12]  5.00  1.57  3.00  3.02  3.92  1.67  3.71  3.50  3.35  4.08  2.75
##  [23]  2.23  7.58  3.18  2.34  2.00  2.00  4.30  3.00  1.45  2.50  3.00
##  [34]  2.45  3.27  3.60  2.00  3.07  2.31  5.00  2.24  2.54  3.06  1.32
##  [45]  5.60  3.00  5.00  6.00  2.05  3.00  2.50  2.60  5.20  1.56  4.34
##  [56]  3.51  3.00  1.50  1.76  6.73  3.21  2.00  1.98  3.76  2.64  3.15
##  [67]  2.47  1.00  2.01  2.09  1.97  3.00  3.14  5.00  2.20  1.25  3.08
##  [78]  4.00  3.00  2.71  3.00  3.40  1.83  5.00  2.03  5.17  2.00  4.00
##  [89]  5.85  3.00  3.00  3.50  1.00  4.30  3.25  4.73  4.00  1.50  3.00
## [100]  1.50  2.50  3.00  2.50  3.48  4.08  1.64  4.06  4.29  3.76  4.00
## [111]  3.00  1.00  4.00  2.55  4.00  3.50  5.07  1.50  1.80  2.92  2.31
## [122]  1.68  2.50  2.00  2.52  4.20  1.48  2.00  2.00  2.18  1.50  2.83
## [133]  1.50  2.00  3.25  1.25  2.00  2.00  2.00  2.75  3.50  6.70  5.00
## [144]  5.00  2.30  1.50  1.36  1.63  1.73  2.00  2.50  2.00  2.74  2.00
## [155]  2.00  5.14  5.00  3.75  2.61  2.00  3.50  2.50  2.00  2.00  3.00
## [166]  3.48  2.24  4.50  1.61  2.00 10.00  3.16  5.15  3.18  4.00  3.11
## [177]  2.00  2.00  4.00  3.55  3.68  5.65  3.50  6.50  3.00  5.00  3.50
## [188]  2.00  3.50  4.00  1.50  4.19  2.56  2.02  4.00  1.44  2.00  5.00
## [199]  2.00  2.00  4.00  2.01  2.00  2.50  4.00  3.23  3.41  3.00  2.03
## [210]  2.23  2.00  5.16  9.00  2.50  6.50  1.10  3.00  1.50  1.44  3.09
## [221]  2.20  3.48  1.92  3.00  1.58  2.50  2.00  3.00  2.72  2.88  2.00
## [232]  3.00  3.39  1.47  3.00  1.25  1.00  1.17  4.67  5.92  2.00  2.00
## [243]  1.75  3.00
```

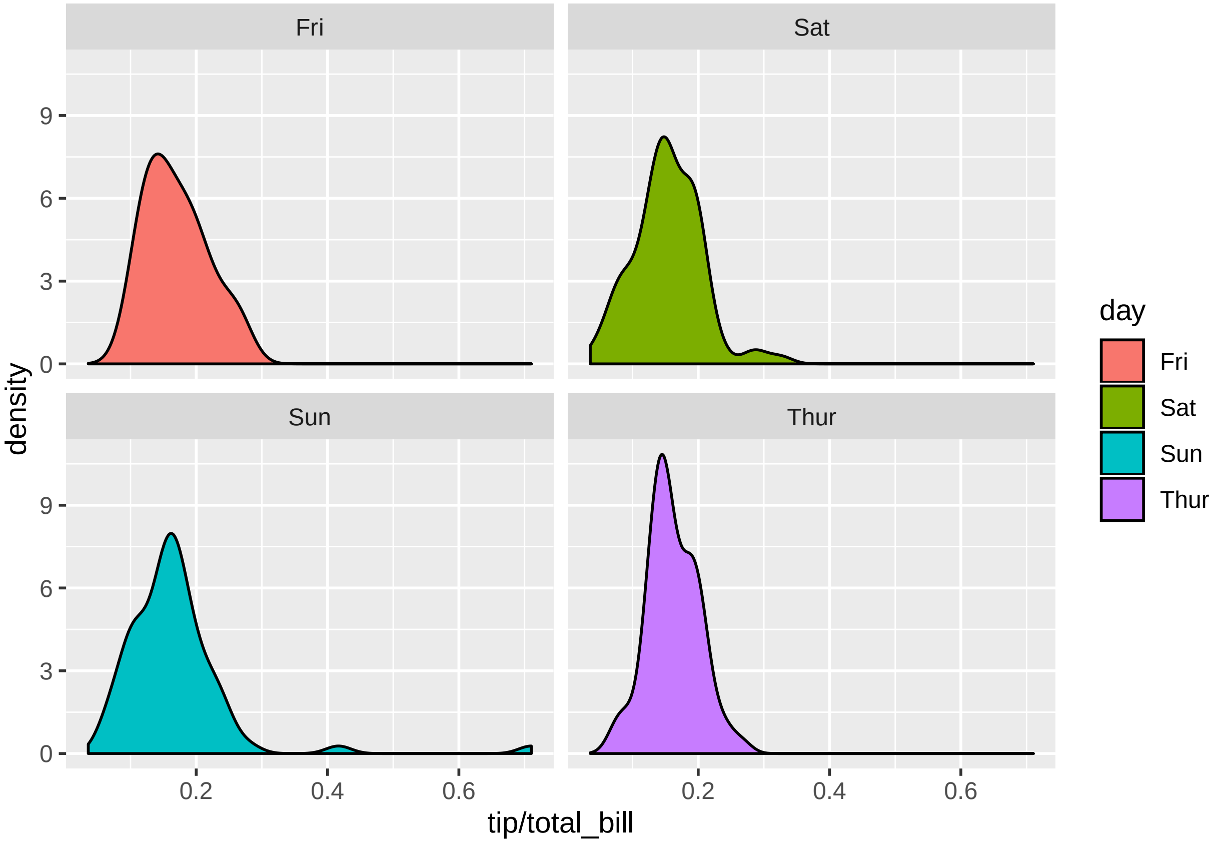
```
ggplot(tips) +
  aes(x = tip) +
  geom_density(
    aes(fill = day)
  )
```



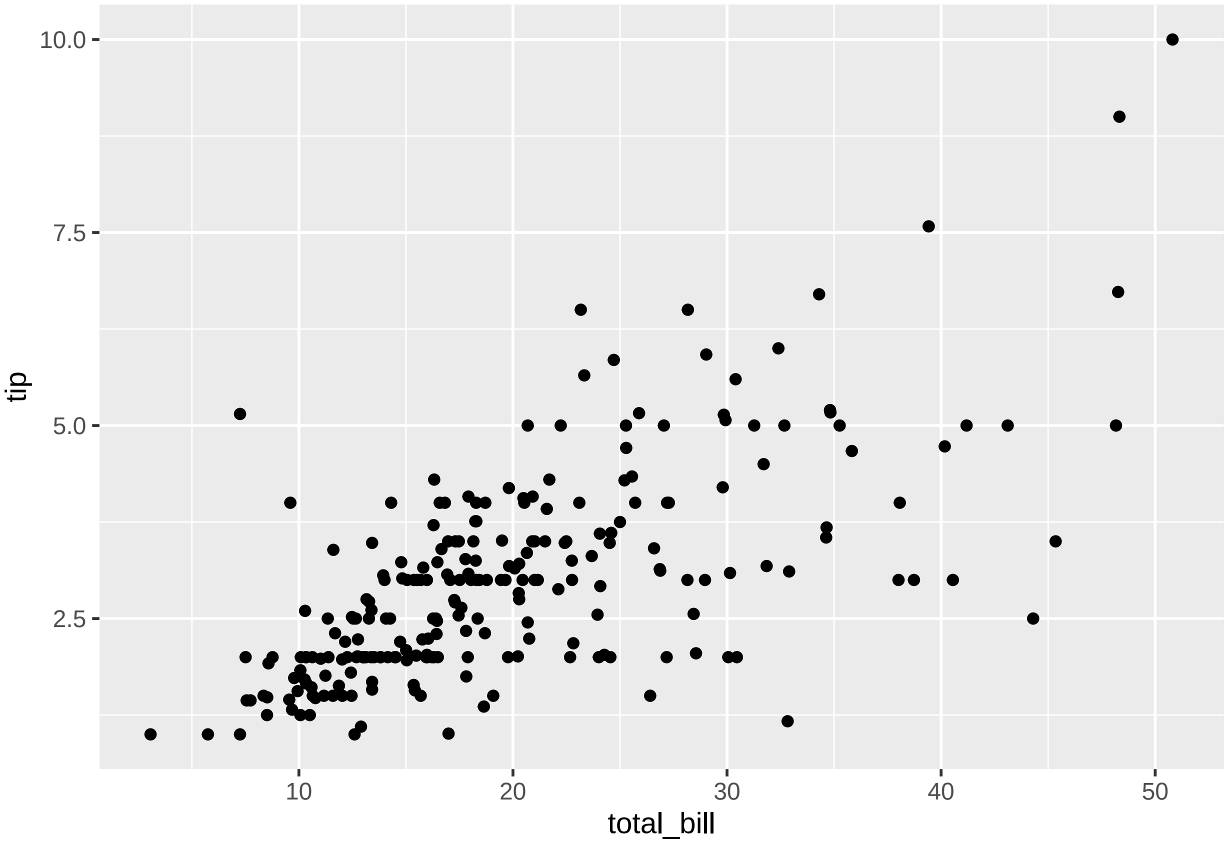
```
ggplot(tips) +
  aes(x = tip) +
  geom_density(
    aes(fill = day),
    alpha = 0.4)
```



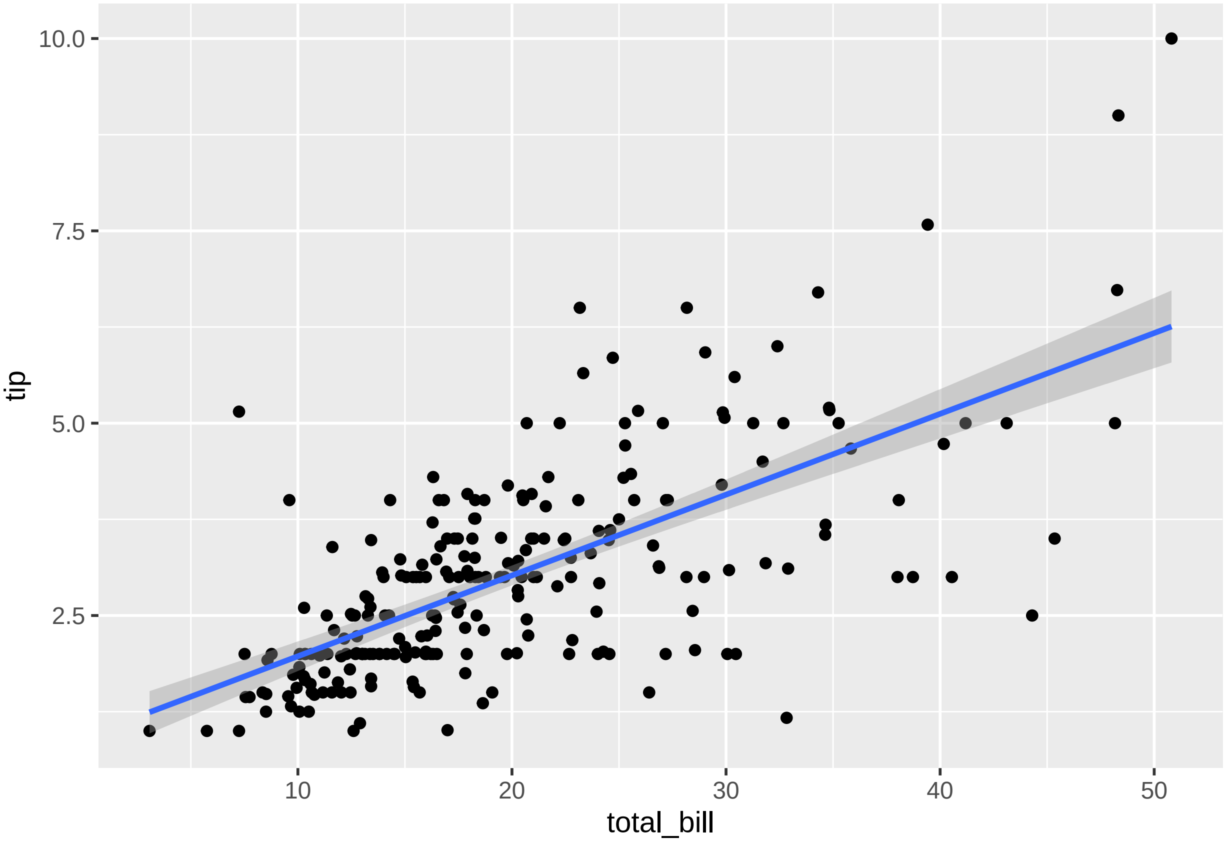
```
ggplot(tips) +
  aes(x = tip/total_bill) +
  geom_density(
    aes(fill = day)
  ) +
  facet_wrap(~ day)
```



```
ggplot(tips) +
  aes(x = total_bill,
      y = tip) +
  geom_point()
```

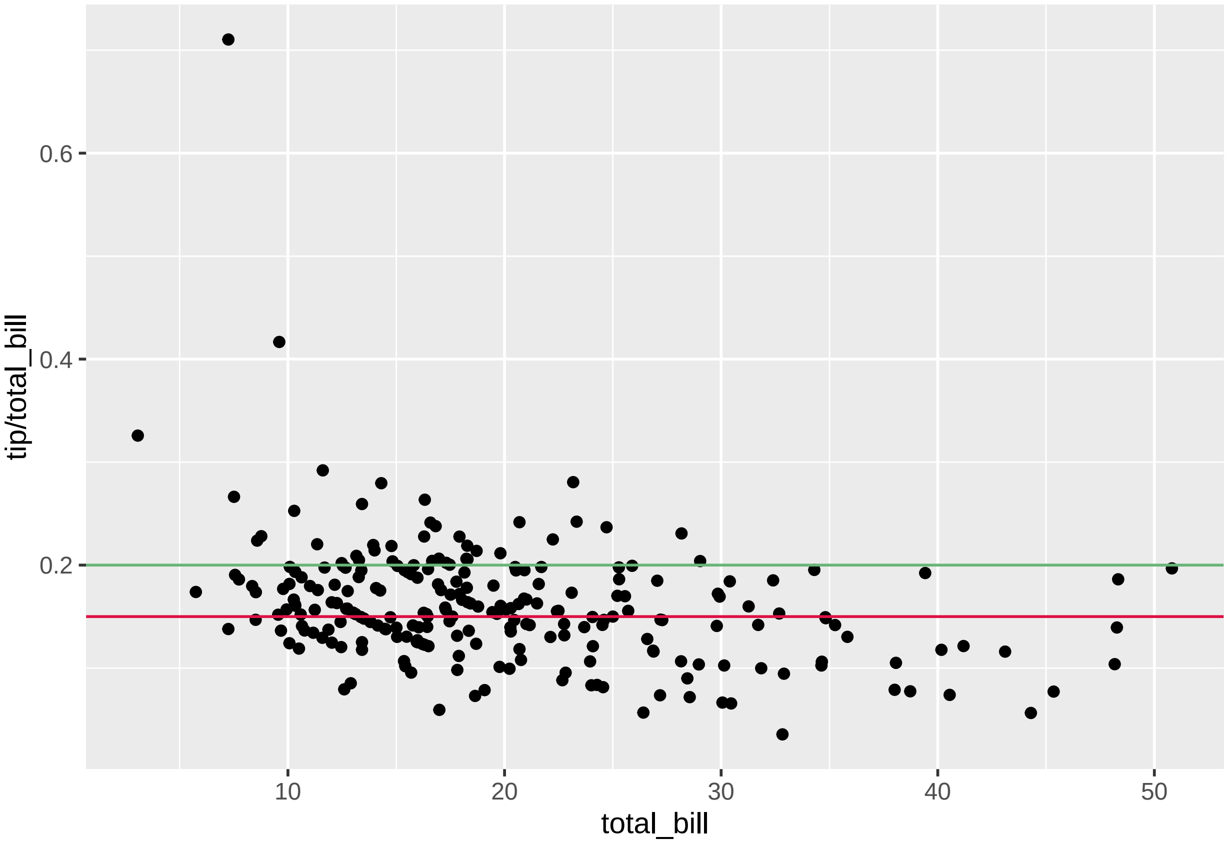


```
ggplot(tips) +
  aes(x = total_bill,
      y = tip) +
  geom_point() +
  geom_smooth(method="lm")
```

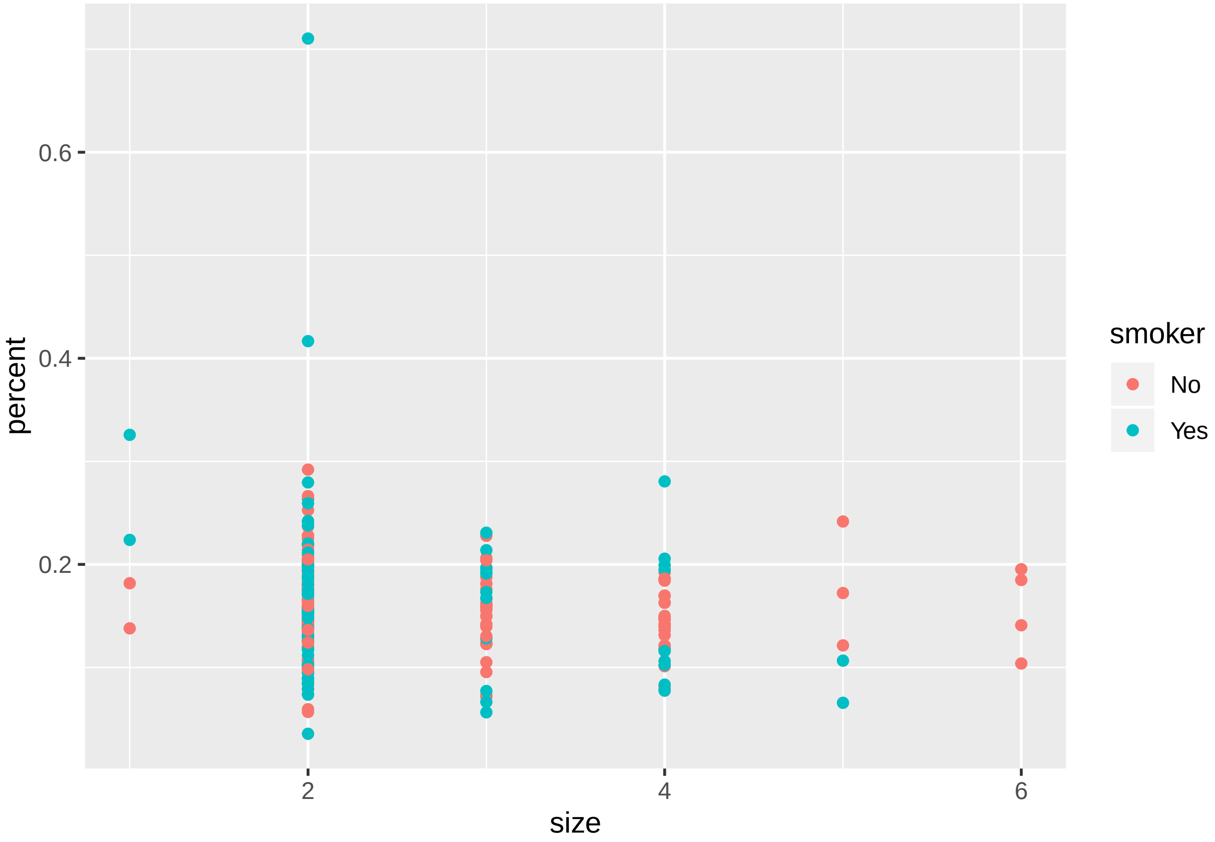


```
graficaPropinas<-ggplot(tips) +
  aes(x = total_bill,
      y = tip,color=day) +
  geom_point() +
  geom_smooth(method = "lm")+
geom_abline(
    slope = c(0.2, 0.15),
    intercept = 0,
    color = c('#69b578',
              "#dd1144"),
    linetype = 3)

ggplotly(graficaPropinas)
```

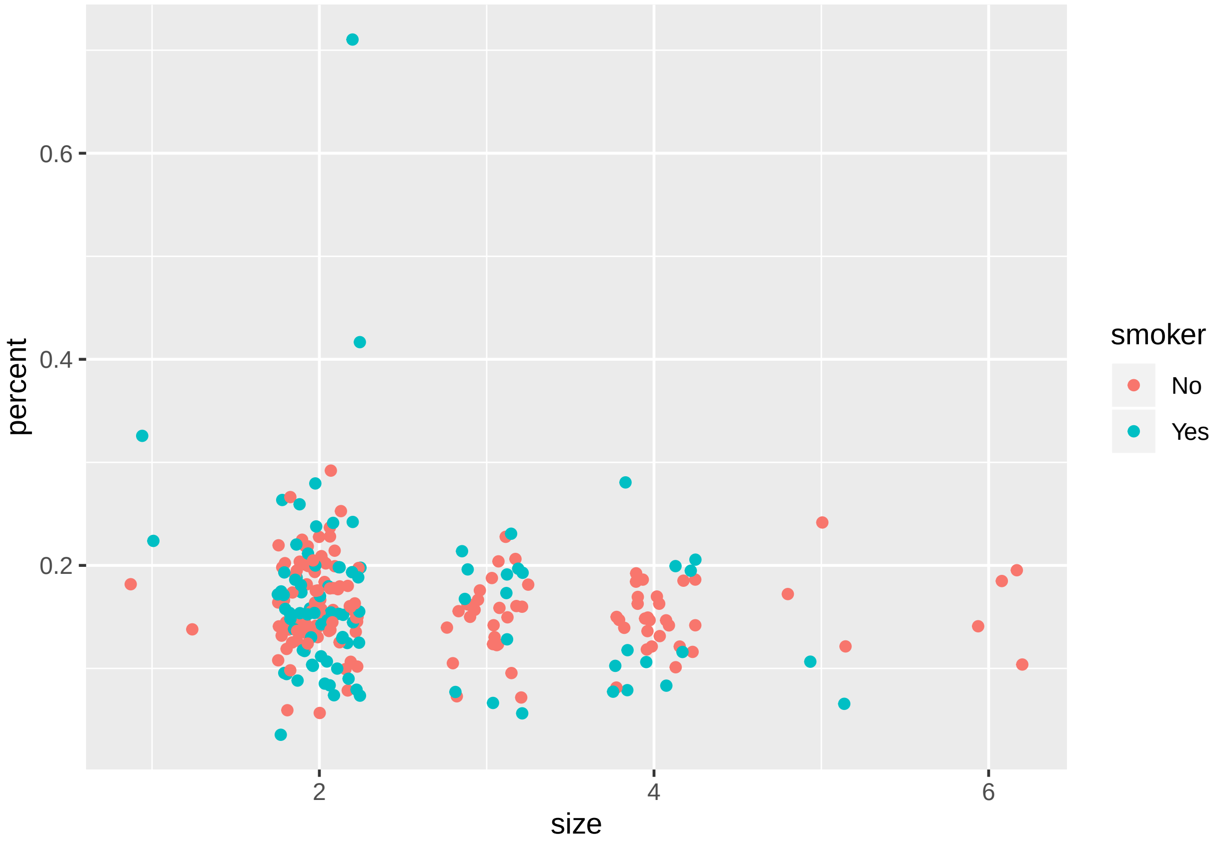
```
ggplot(tips) +
  aes(x = total_bill,
      y = tip/total_bill) +
  geom_point() +
  geom_hline(
    yintercept = c(0.2, 0.15),
    color = c('#69b578',
              "#dd1144"),
    linetype = 1)
```



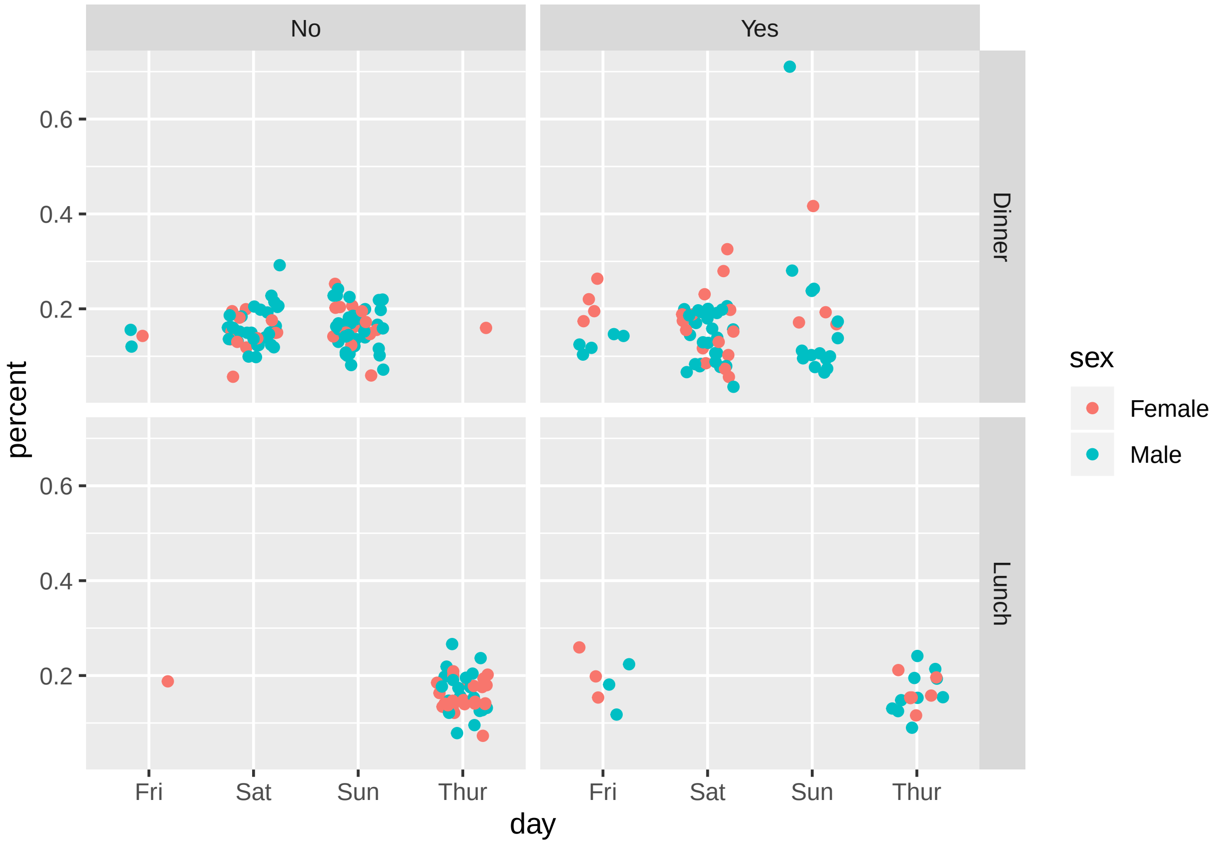
```
tips$percent <-
  tips$tip/tips$total_bill
ggplot(tips) +
  aes(x = size,
      y = percent,
      color = smoker) +
  geom_point()
```



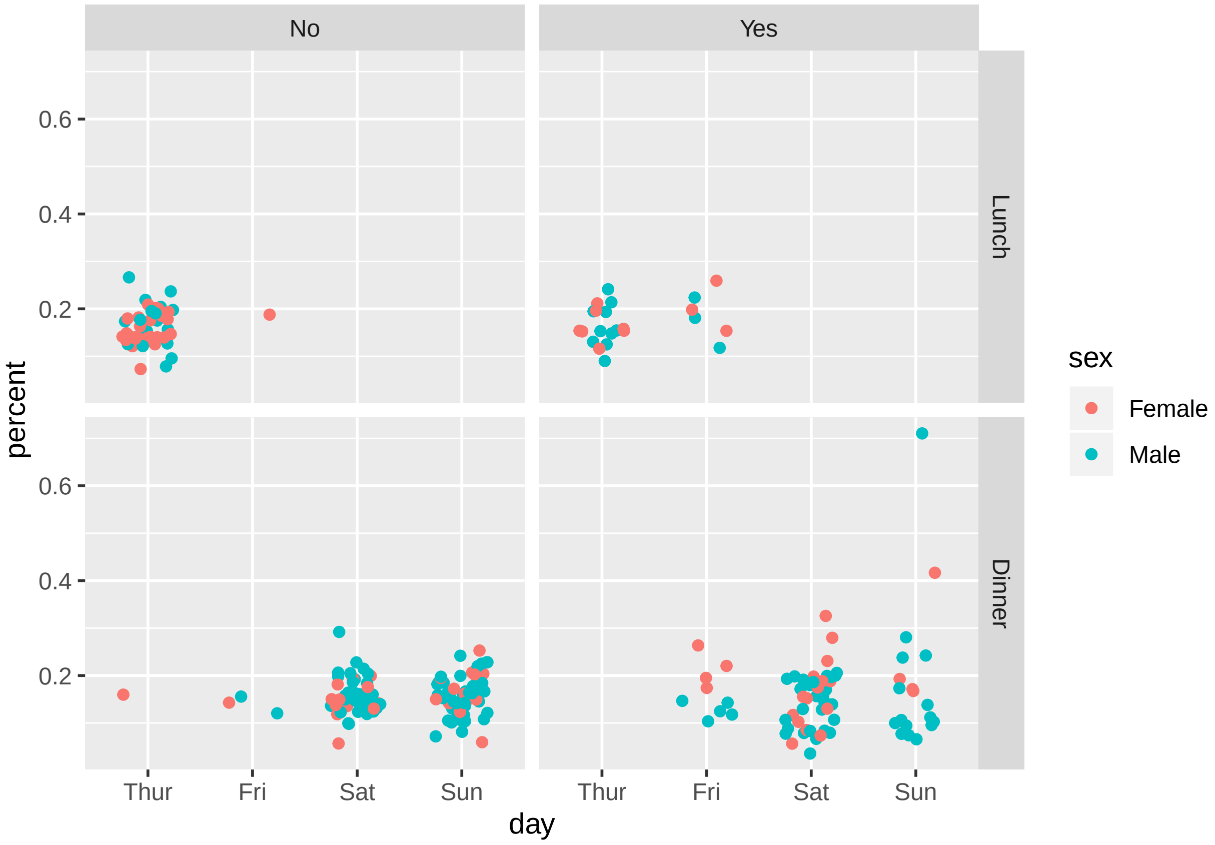
```
tips$percent <-
  tips$tip/tips$total_bill
ggplot(tips) +
  aes(x = size,
      y = percent,
      color = smoker) +
  geom_jitter(width = 0.25)
```



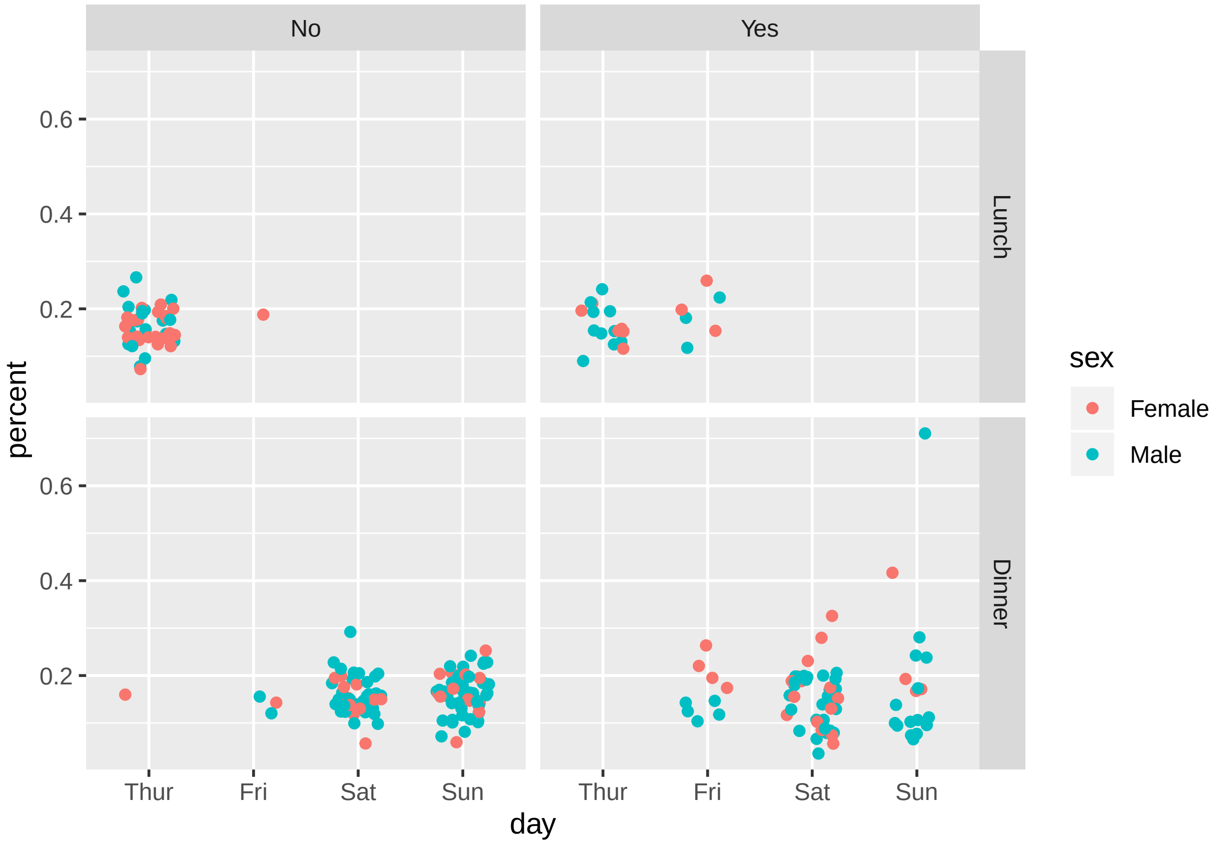
```
ggplot(tips) +
  aes(x = day,
      y = percent,
      color = sex) +
  geom_jitter(width = 0.25) +
  facet_grid(time ~ smoker)
```



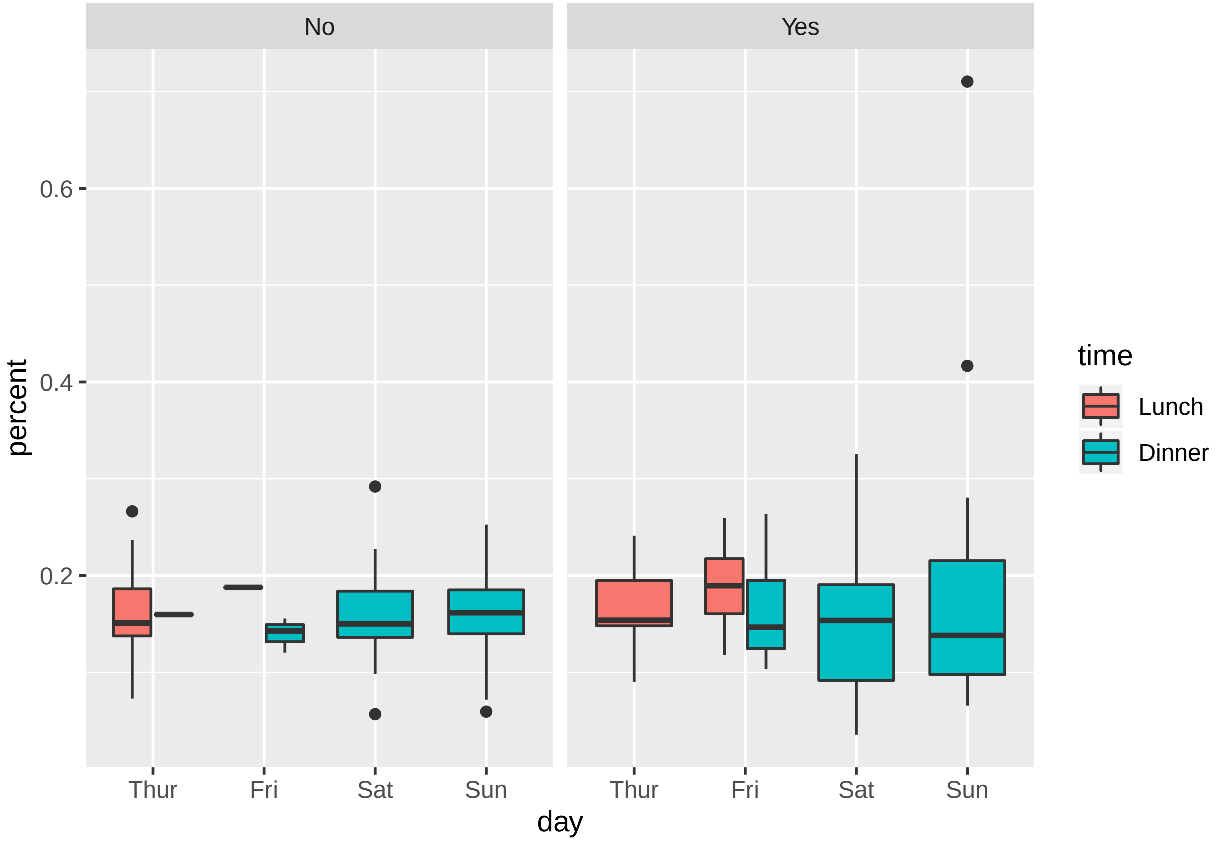
```
tips <- mutate(tips,
  time = factor(time,
    c("Lunch", "Dinner")),
  day = factor(day,
    c("Thur", "Fri",
      "Sat", "Sun")
  ))
ggplot(tips) +
  aes(x = day,
      y = percent,
      color = sex) +
  geom_jitter(width = 0.25) +
  facet_grid(time ~ smoker)
```



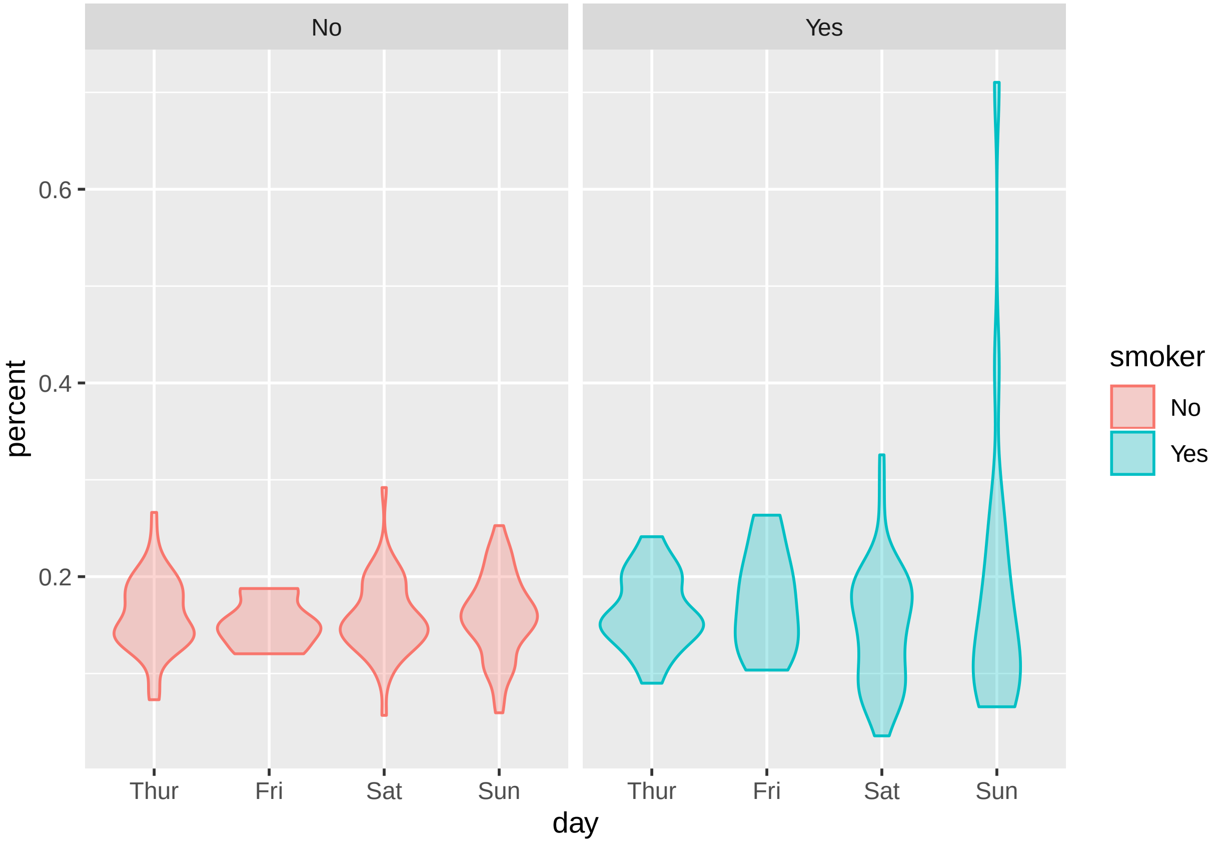
```
ggplot(tips) +
  aes(x = day,
      y = percent,
      color = sex) +
  geom_jitter(width = 0.25) +
  facet_grid(time ~ smoker)
```



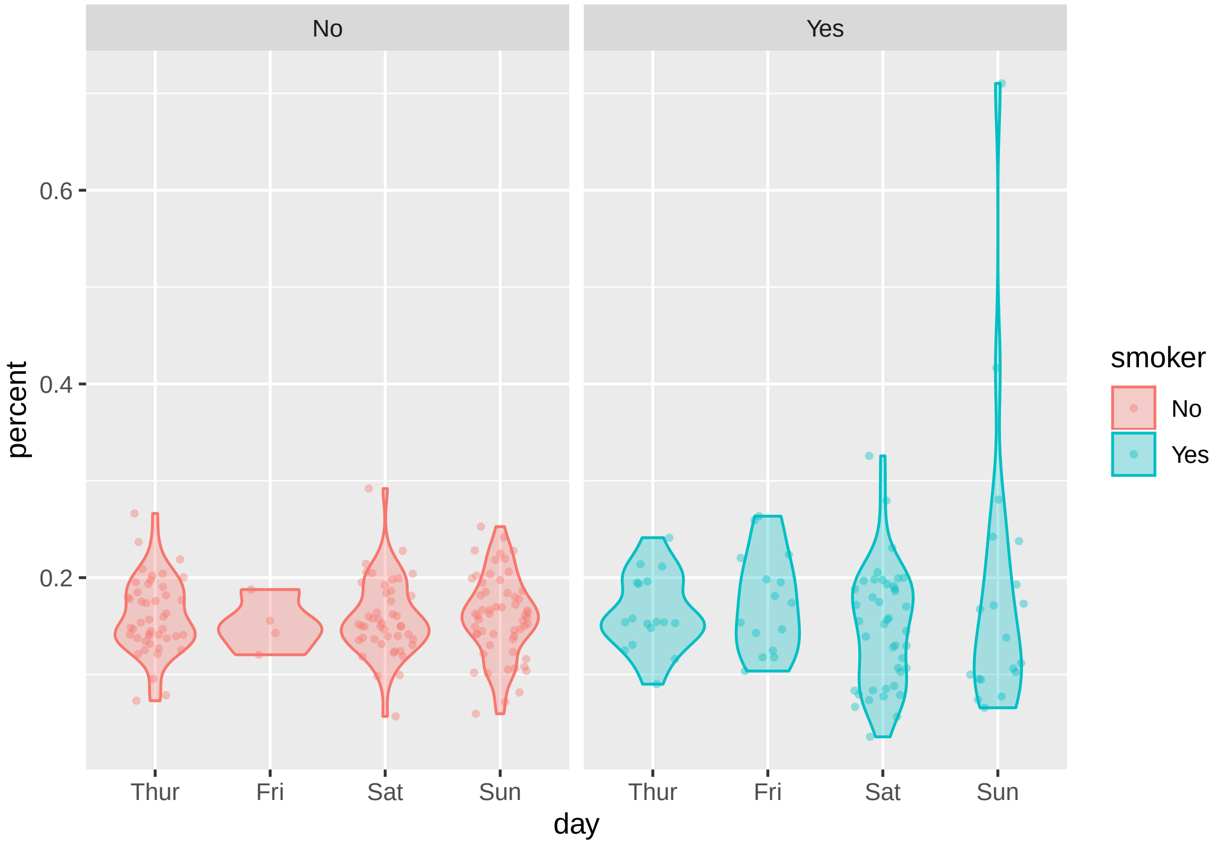
```
ggplot(tips) +
  aes(x = day,
      y = percent,
      fill = time) +
  geom_boxplot() +
  facet_grid(. ~ smoker)
```



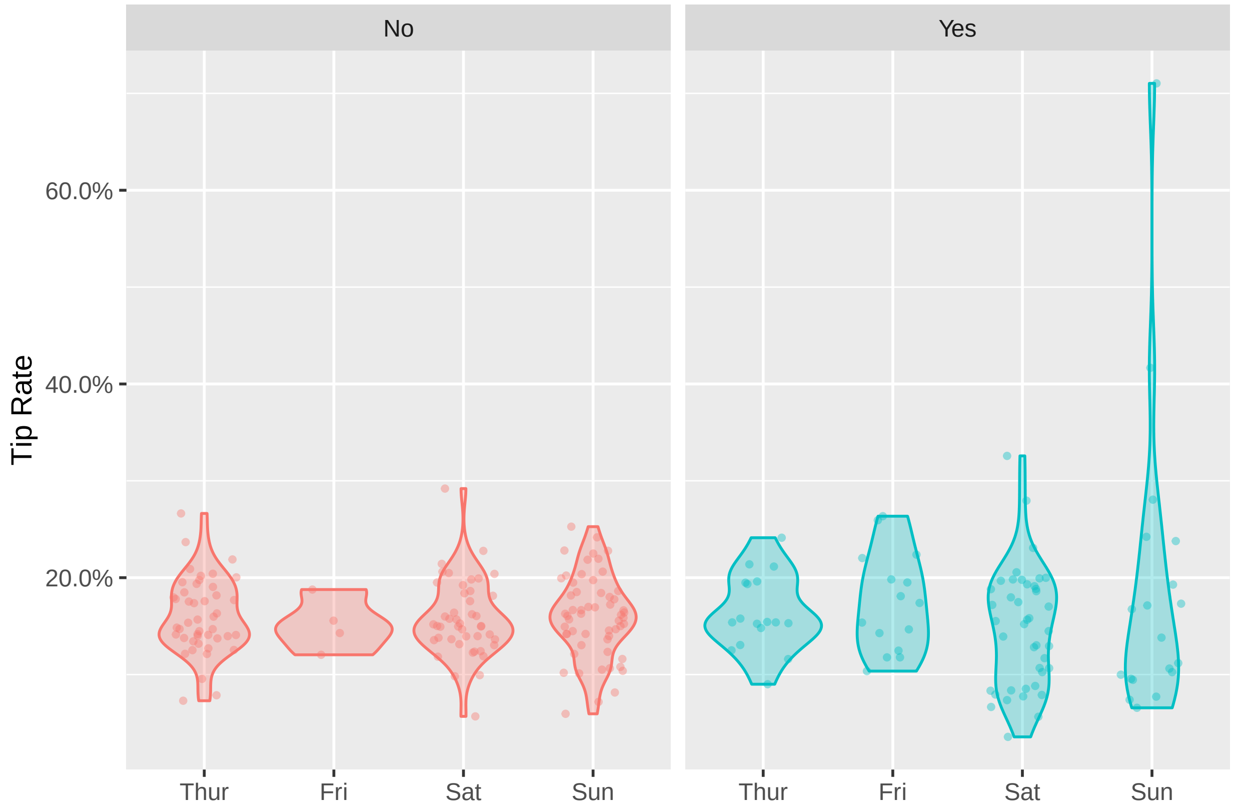
```
ggplot(tips) +
  aes(x = day,
      y = percent,
      color = smoker,
      fill  = smoker) +
  geom_violin(alpha = 0.3) +
  facet_wrap(~ smoker)
```



```
g <- ggplot(tips) +
  aes(x = day,
      y = percent,
      color = smoker,
      fill = smoker) +
  geom_violin(alpha = 0.3) +
  geom_jitter(alpha = 0.4,
              width = 0.25,
              size  = 0.8)+
  facet_wrap(~ smoker)
g
```



```
g + guides(color = FALSE,
           fill  = FALSE) +
  labs(x = '',
       y = 'Tip Rate') +
  scale_y_continuous(
    labels = scales::percent
  )
```



Additional Resources

- R for Data Science: http://r4ds.had.co.nz/