

## Visualizaciones en ggplot2

### Reto de la sección

Te han contactado como analista de datos por una página web de críticas de películas. Están escribiendo un artículo donde analizan los ratings de los críticos y de la audiencia, así como los presupuestos para las películas de los años 2007-2011.

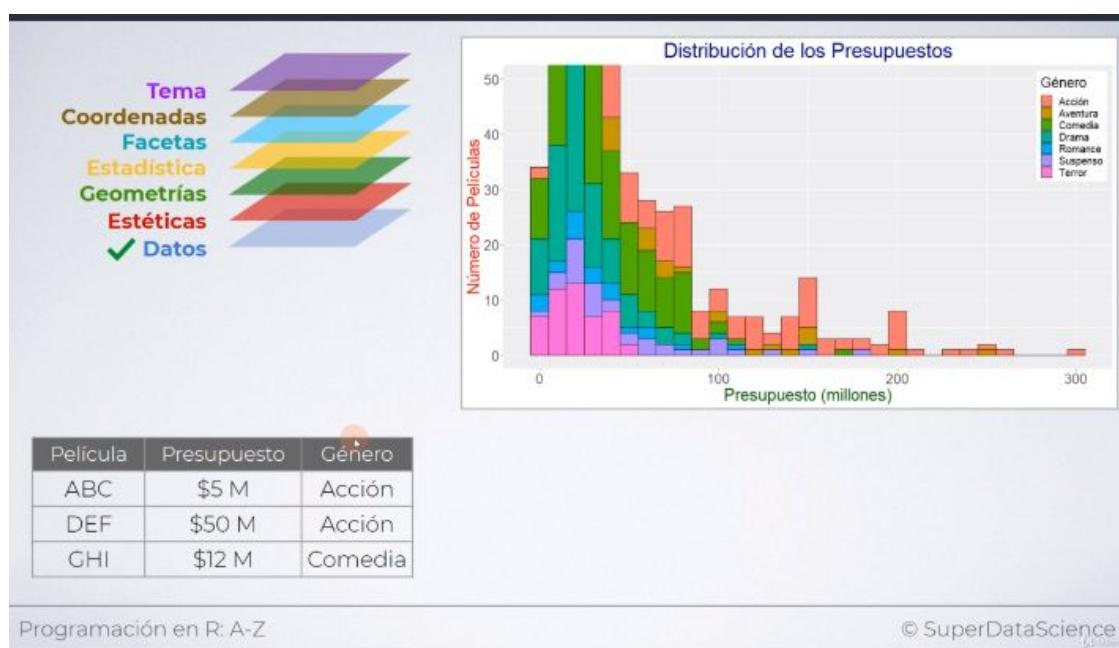
Esta es la primera vez que están haciendo este análisis y no saben exactamente qué necesitan. Te han pedido que veas los datos y que les generes 5 gráficas que cuenten una historia de los datos.

Sin embargo, hay un gráfico que el CEO pidió específicamente - un diagrama que muestre cómo ha ido evolucionando a través de los años la correlación entre los ratings de la audiencia y de los críticos, por género. Esto es adicional a los otros 5 gráficos.

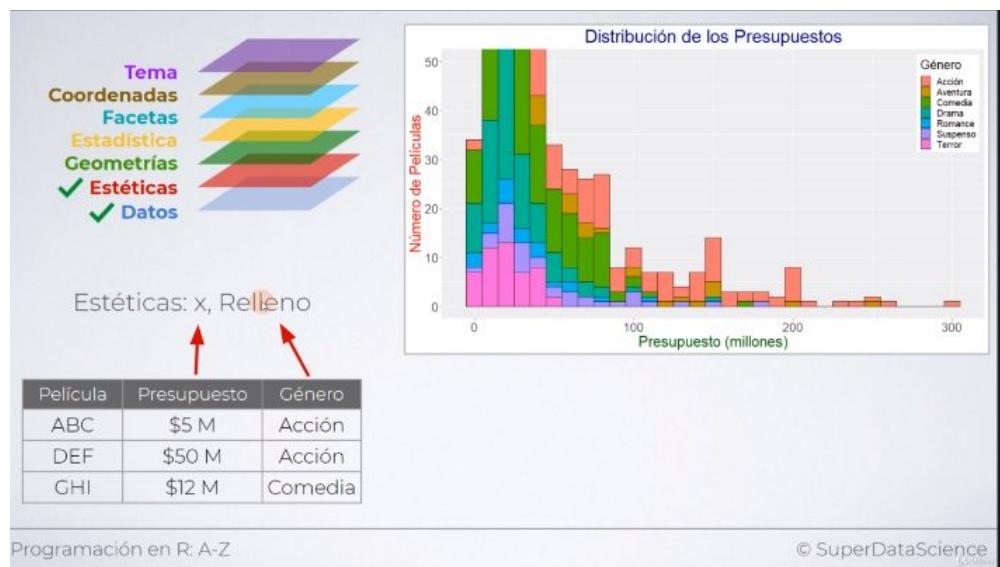
# Gramática de Datos



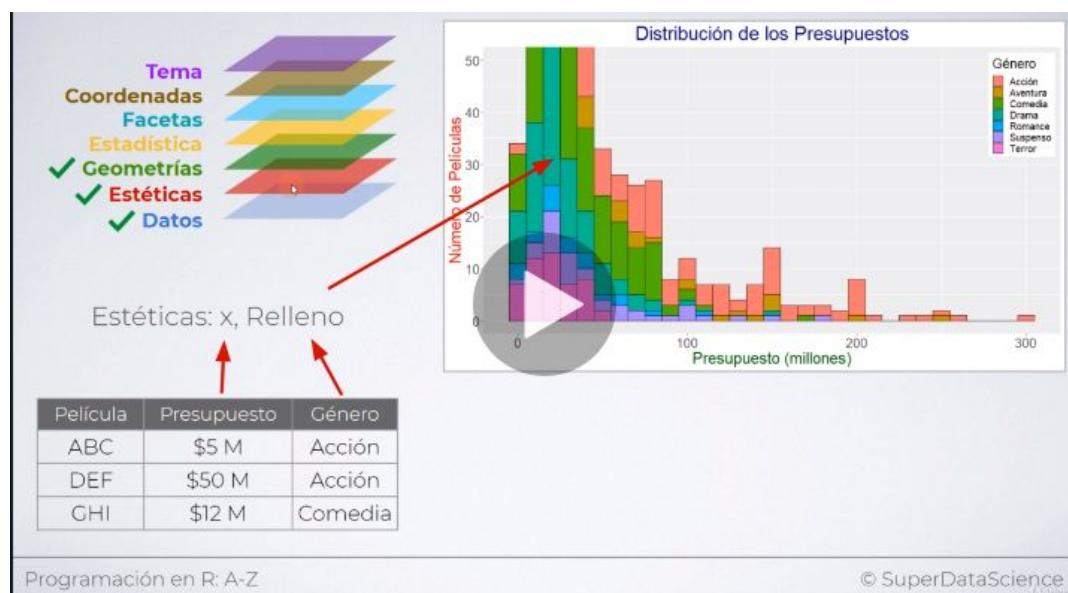
## • Datos



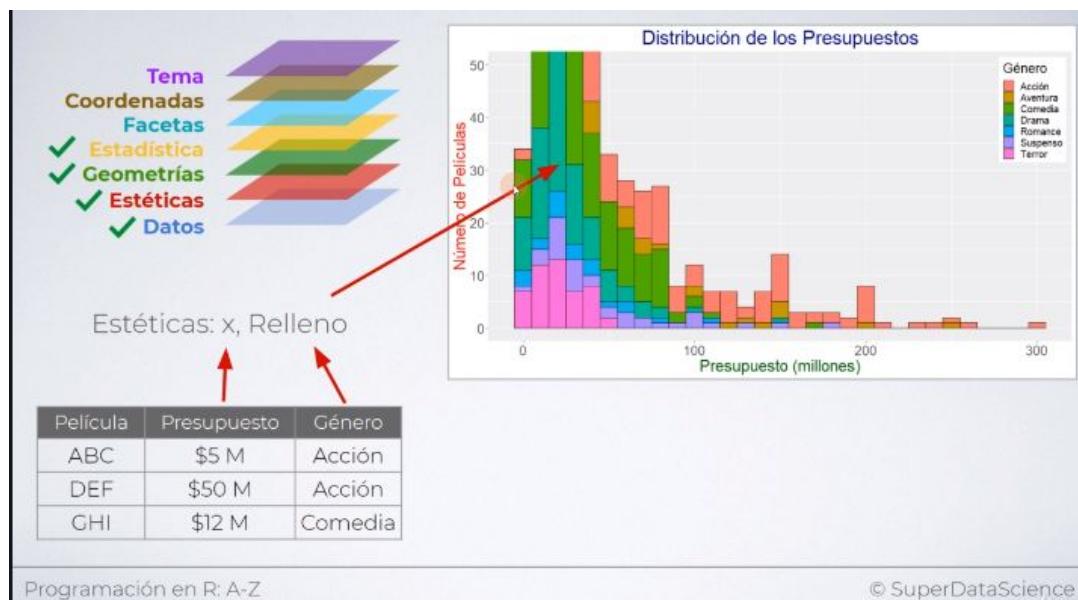
- **Estéticas** es un elemento representado en el gráfico, no es el número propiamente sino la relación de este dato a algo como el color, el tamaño, etc.



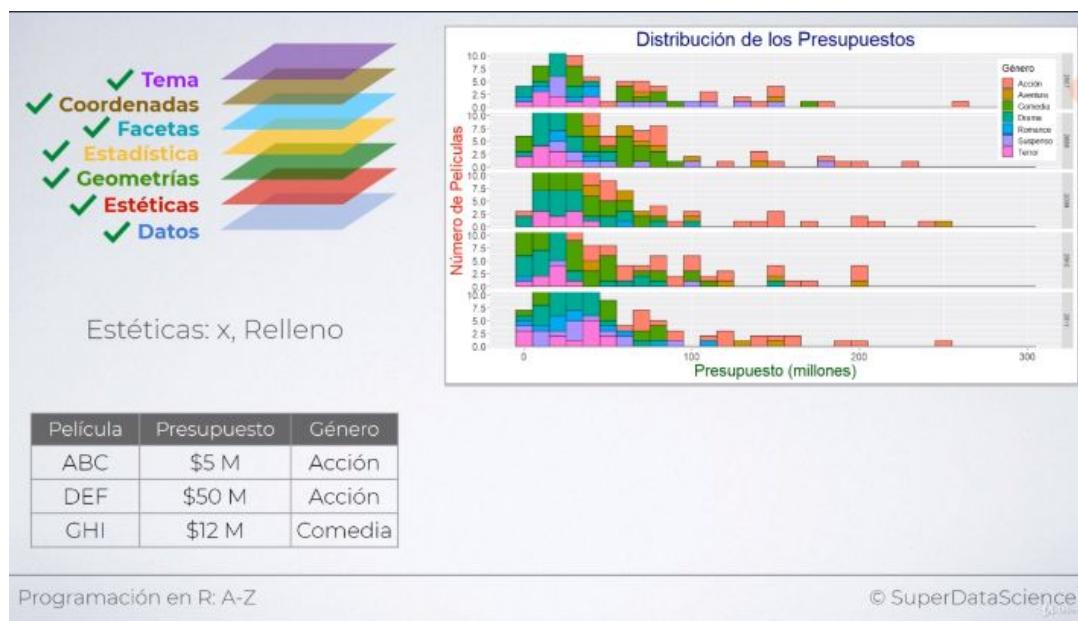
- **Geometría** define la forma de la capa anterior, algo así como el boxplot, diagramas de tallos, gráficos de dispersión, barras, pie, etc.



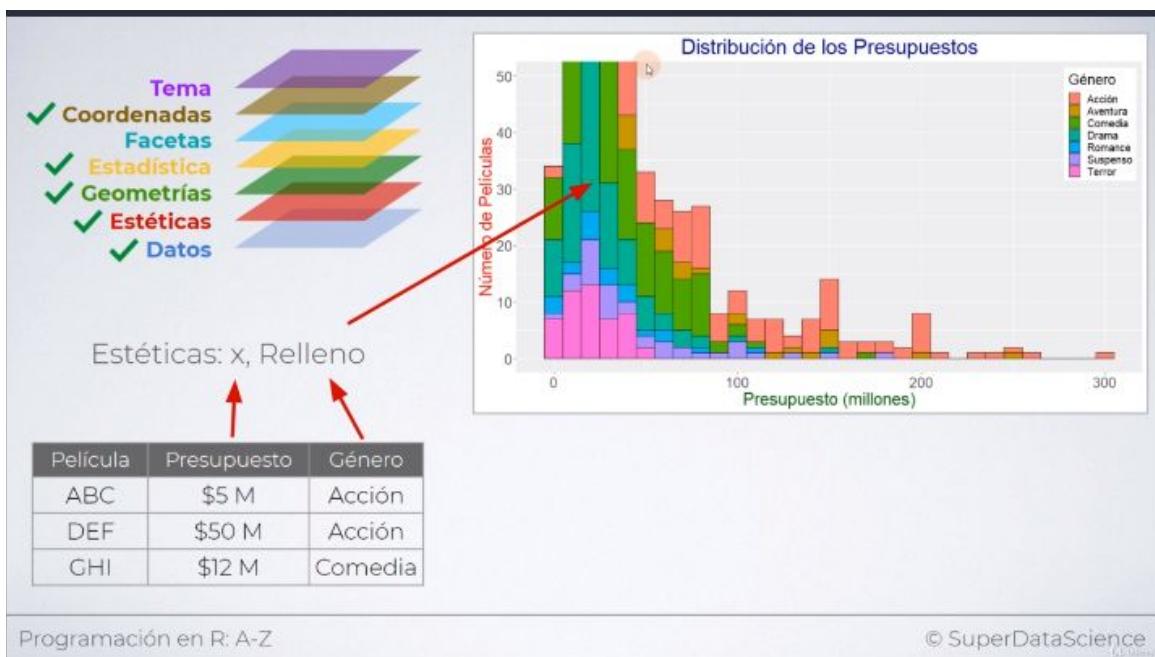
- **Estadística**, es la capa de manipulación de los datos por medio de modelos estadísticos que permite nuevas formas de comprender los datos. En el ejemplo de abajo vemos esto en el mismo histograma que genera una distribución de los datos, la cual no es normal sin sesgada.



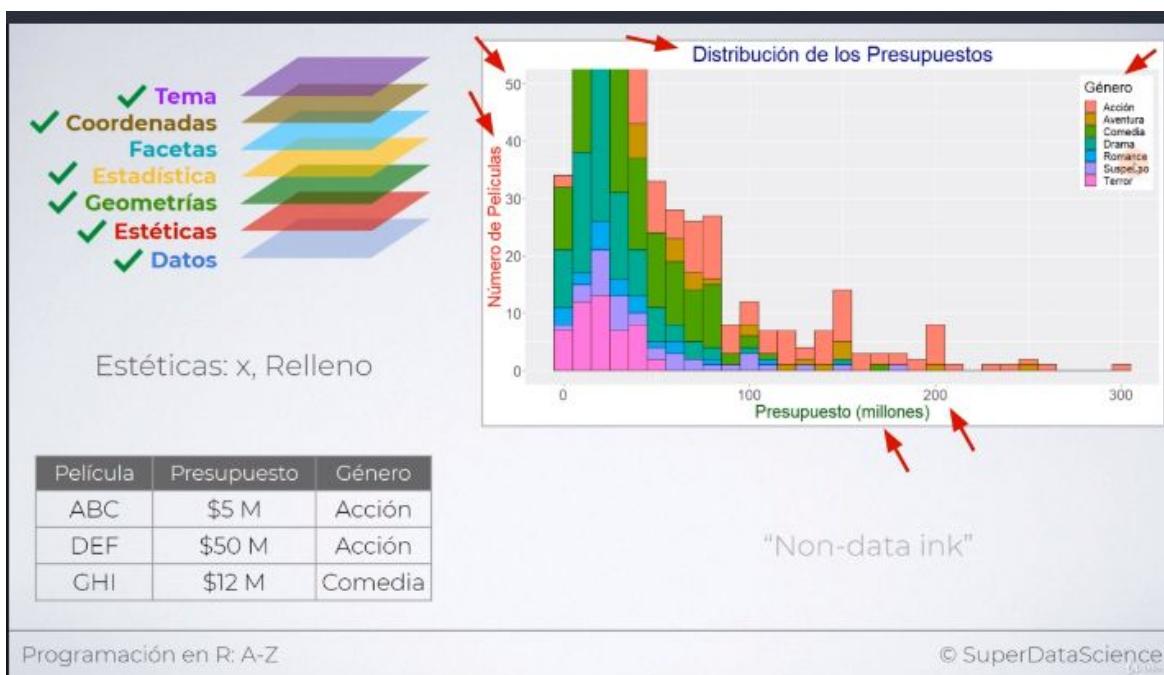
- **Facetas**, son cuando un gráfico X se divide de N maneras dada N categorías distintas; como los años el el ejemplo de abajo.



- **Coordenadas**, ya sean (x,y), polares, (x,y,z), etc. En el ejemplo de abajo se ve en los ejes (x, y) y en el zoom que tienen los datos



- **Tema**, el resto del gráfico que no está asociado con los datos en manera directa. En el caso de abajo tenemos nombres, letras, título, etc.



## Factores (capa datos)

```
#Primero seleccionamos nuestros datos  
datos.peliculas <- read.csv(file.choose())  
head(datos.peliculas)
```

```
#Revisamos los datos  
colnames(datos.peliculas) <- c("Pelicula", "Genero", "RatingCriticos", "RatingAudienicia",  
"PresupuestoMillones", "Año")  
head(datos.peliculas)  
tail(datos.peliculas)  
str(datos.peliculas)
```

#Esto no es del curso, pero fue necesario abrir el paquete **dplyr**, un editor de gramática de datos, para así poder poner como factores los datos character del documento proporcionado.  
#Lo que hace es usar la función **mutate\_at()** para poder cambiar el tipo de dato de character a factor.

```
library(dplyr)  
datos.peliculas <- mutate_at(datos.peliculas, vars(Genero), as.factor)  
datos.peliculas <- mutate_at(datos.peliculas, vars(Pelicula), as.factor)  
str(datos.peliculas)  
summary(datos.peliculas)
```

#Ahora al parecer nos están enseñando cómo alterar esto pero sin necesidad del tidyverse.

```
#Esto mediante factor()  
factor(datos.peliculas$Año)  
datos.peliculas$Año <- factor(datos.peliculas$Año)  
summary(datos.peliculas)  
str(datos.peliculas)
```

#Por esto anterior, podemos ver que pudimos haber hecho lo siguiente de en lugar de alterar los datos mediante dplyr

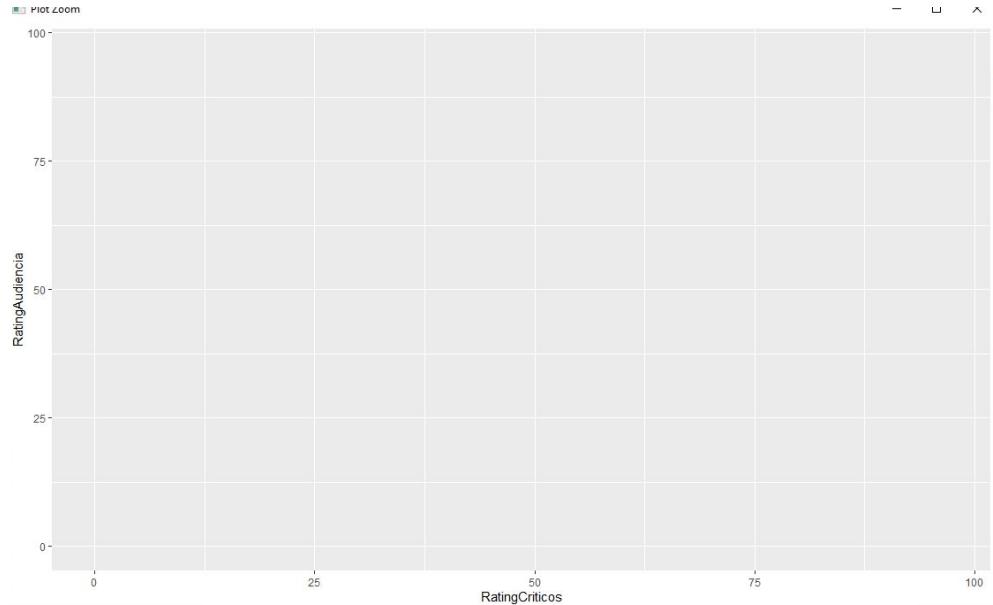
```
datos.peliculas$Año <- factor(datos.peliculas$Año)
```

## Estéticas y Geometrías

#Estéticas

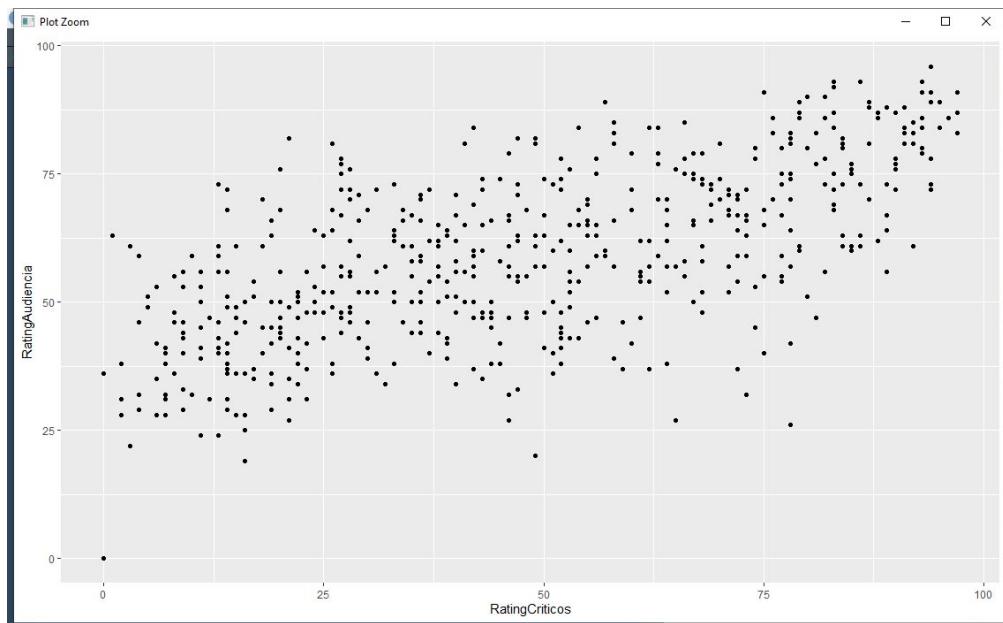
```
library(ggplot2)  
ggplot(data=datos.peliculas, aes(x=RatingCriticos, y=RatingAudien  
cia))
```

#Al aplicar esto anterior solamente se obtiene el eje (x,y), pero no obtenemos nada más  
#Esto se debe a que las estéticas solo van a generar la relación entre los datos y el gráfico  
pero no el gráfico en sí. Para esto se necesitan geometrías



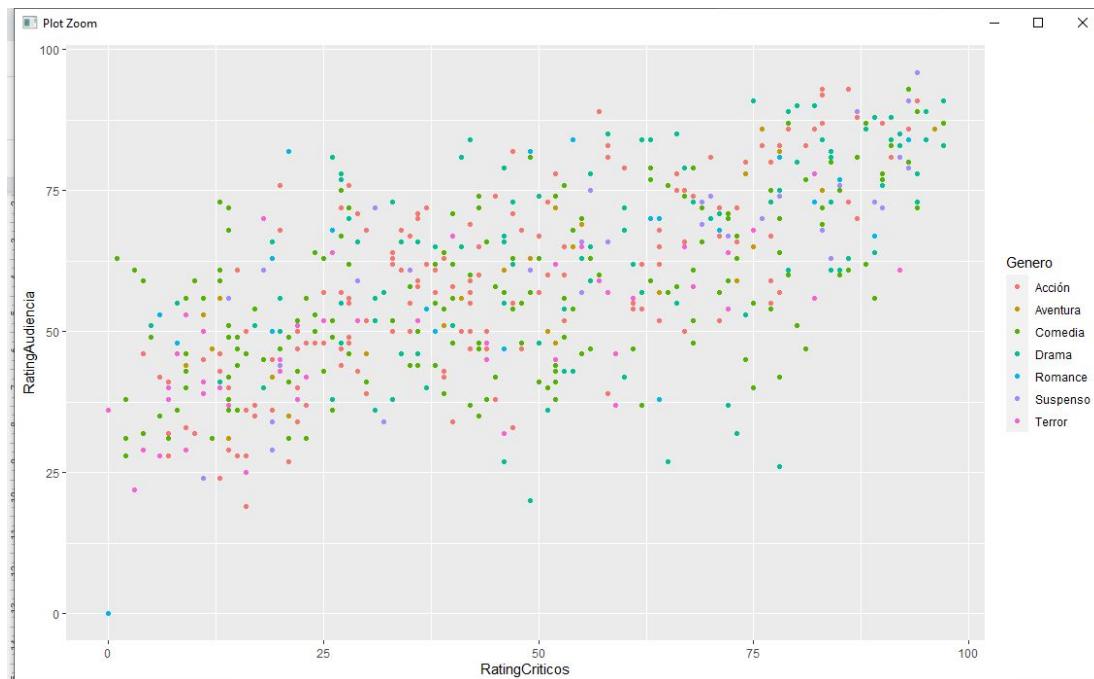
#Geometrías

```
ggplot(data=datos.peliculas, aes(x=RatingCriticos, y=RatingAudien  
cia))+  
  geom_point()
```



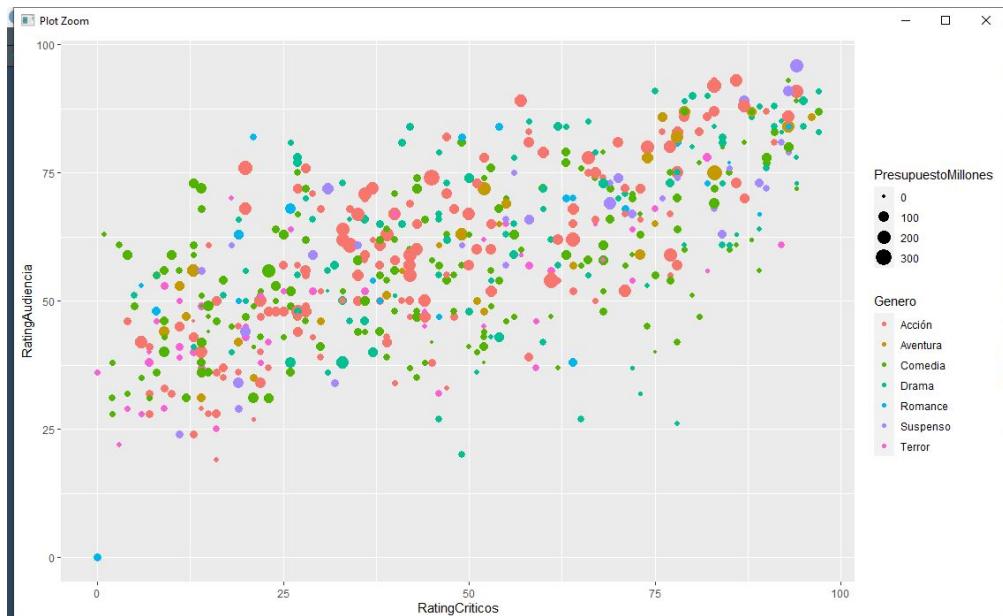
```
#Para agregar color usamos aes()
```

```
ggplot(data=datos.peliculas, aes(x=RatingCriticos, y=RatingAudencia, color=Genero))+  
  geom_point()
```



#Para poder alterar el tamaño de los puntos

```
ggplot(data=datos.peliculas, aes(x=RatingCriticos, y=RatingAudiencia, color=Genero, size=PresupuestoMillones))+  
  geom_point()
```



## Graficando con Capas (geometrías)

#Graficando con Capas

#Primero vamos a convertir nuestro código de ggplot en un objeto

```
A <- ggplot(data=datos.peliculas, aes(x=RatingCriticos, y=RatingAudien  
cia, color=Genero,  
size= PresupuestoMillones))
```

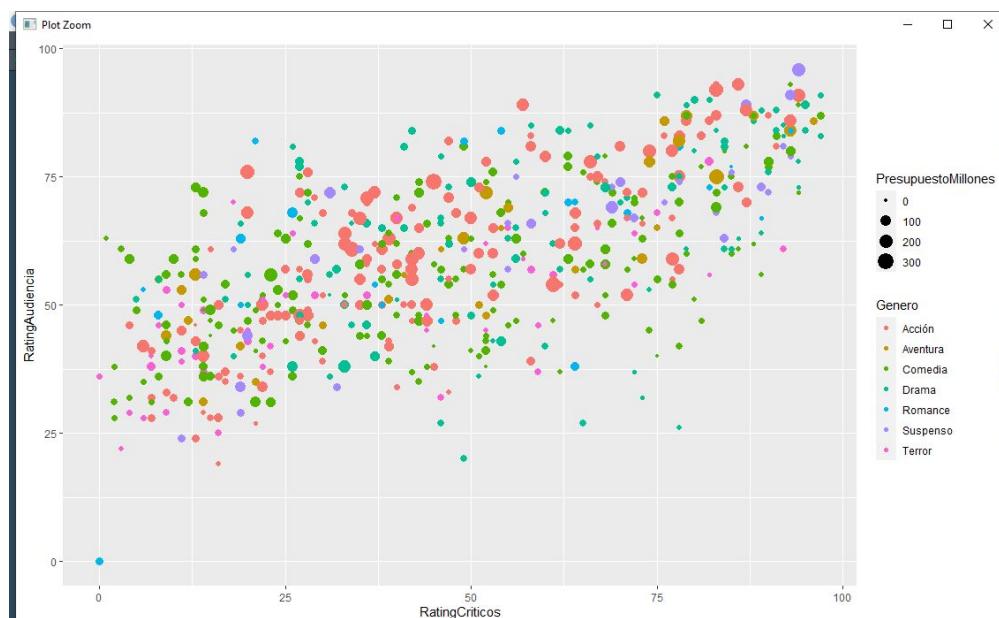
#Si corremos A tendremos solamente un gráfico sin puntos

```
A
```

#Así tendremos de nuevo el último gráfico de la sección anterior

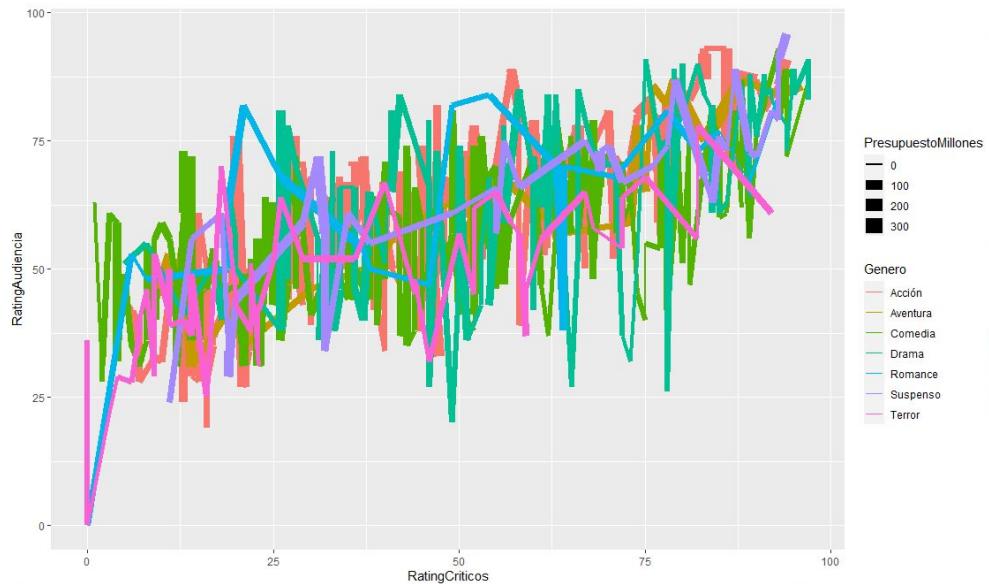
**#Estas son las capas, primero el objeto que corresponde al ggplot( estética) y luego una  
capa de tipo geométrica con geom\_point()**

```
A + geom_point()
```



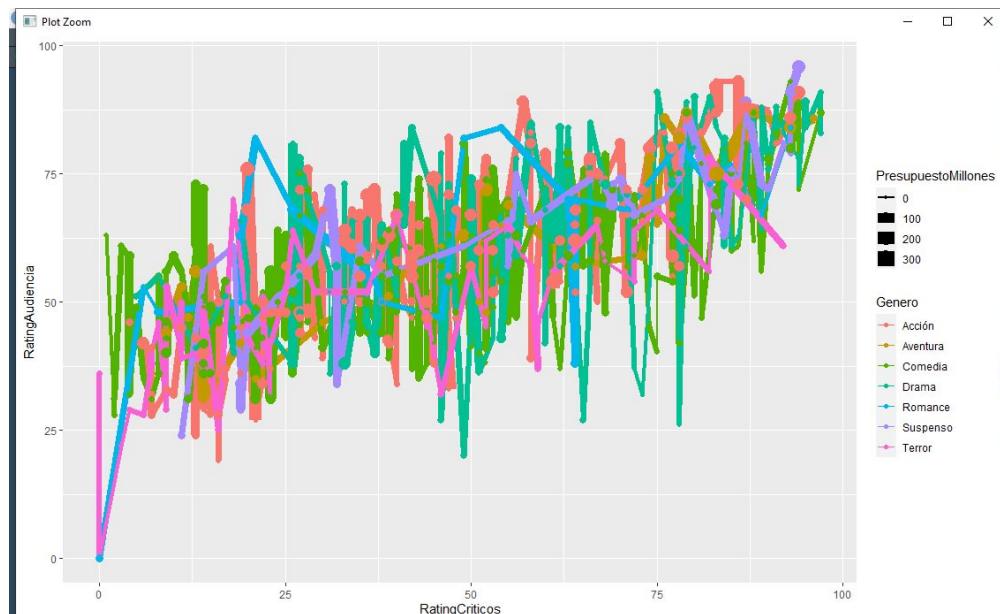
#Lo mismo pero con líneas

```
A + geom_line()
```



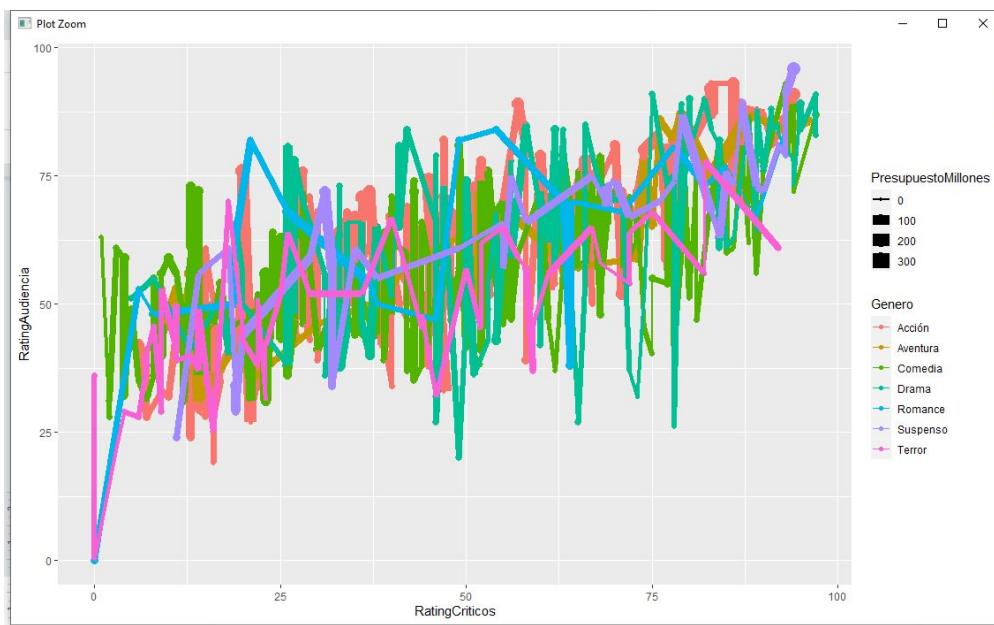
#Podemos combinar las capas, en este caso las línea y los puntos

A + geom\_line() + geom\_point()



#El orden de las capas es importante como lo ilustra la siguiente gráfica donde no se ven los puntos por ponerlos primero

A + geom\_point() + geom\_line()



## Sobreescribiendo estéticas

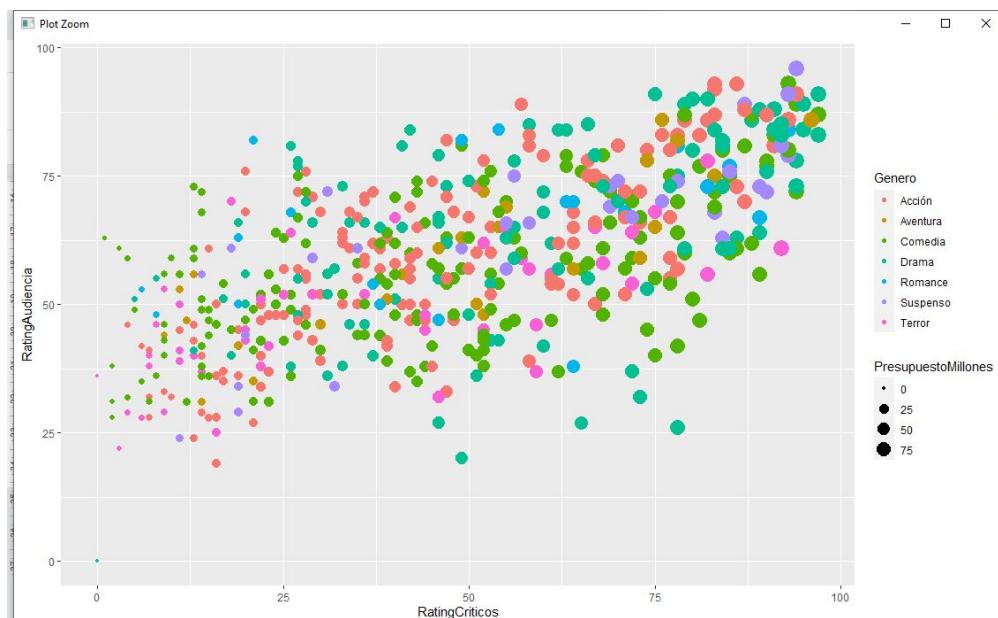
```
#Sobreescribiendo estéticas
```

```
B <- ggplot(data= datos.peliculas, aes(x = RatingCriticos, y = RatingAudien  
cia,  
size= PresupuestoMillones, color=Genero))  
B + geom_point()
```

```
#Debido a que geom_point es una capa sobre el objeto, nosotros podemos sobreescibir las  
estéticas derivadas del objeto
```

```
#Caso 1
```

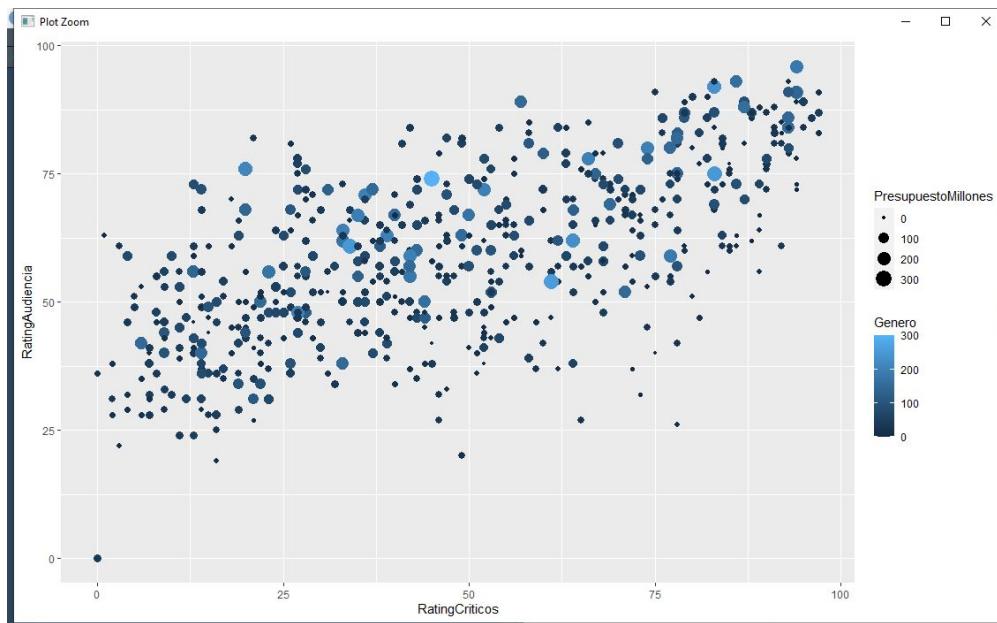
```
B + geom_point(aes(size=RatingCriticos))
```



```
#Caso 2
```

```
B + geom_point(aes(color=PresupuestoMillones))
```

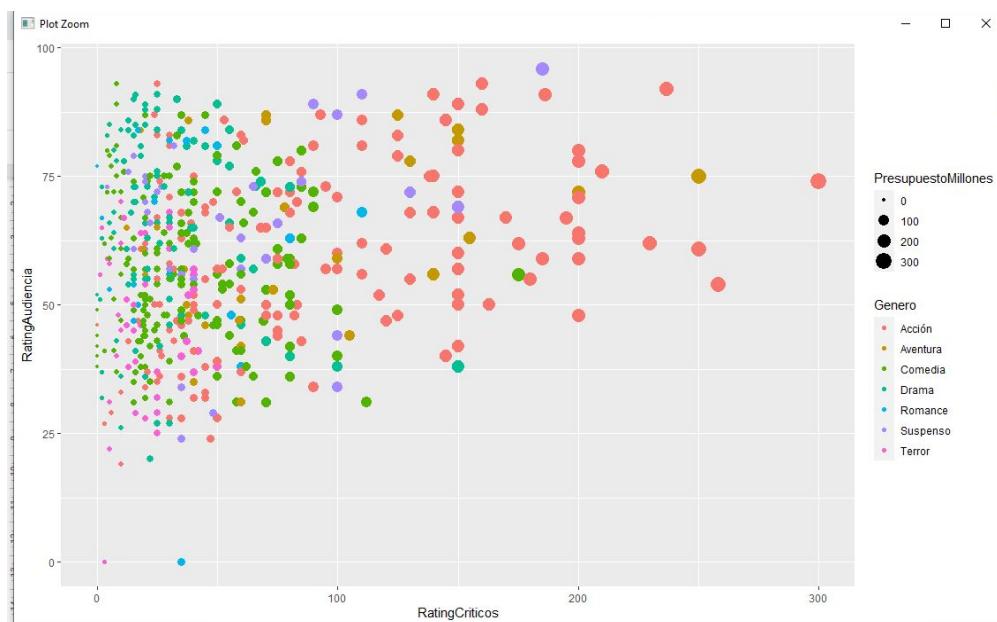
```
#Como se ve, la etiqueta de género que pertenecía a B se mantiene aunque hayamos  
cambiado el color en la capa. Esto es porque el objeto se mantiene y como una capa de  
pintura esta no altera al objeto
```



#Este mismo fenómeno se observa en el siguiente ejemplo, nuestra capa original de estética se mantendrá.

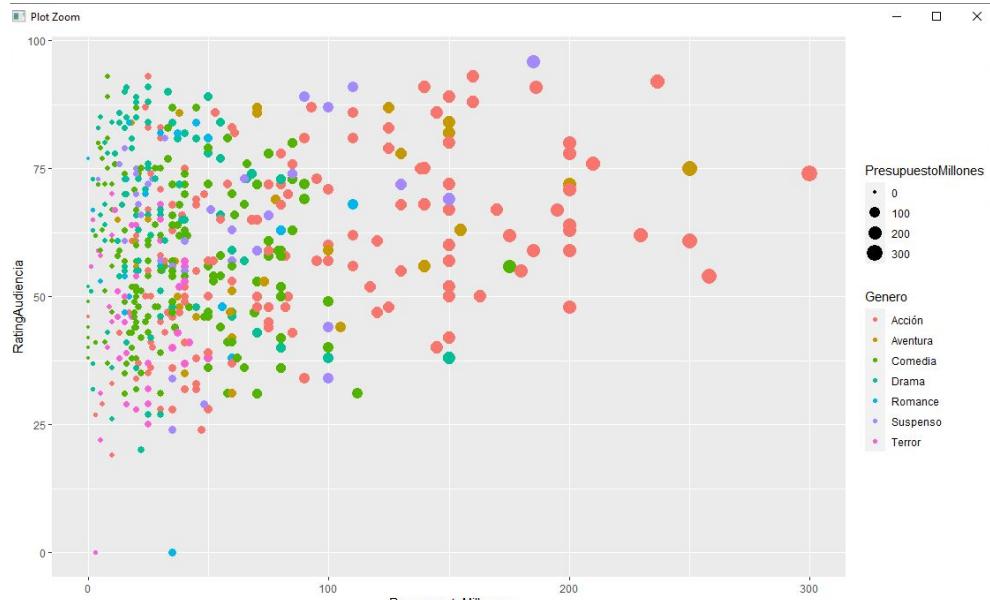
#Se mantuvieron los RatingCriticos, a pesar de que la geometría dice lo contrario

```
B + geom_point(aes(x= PresupuestoMillones))
```



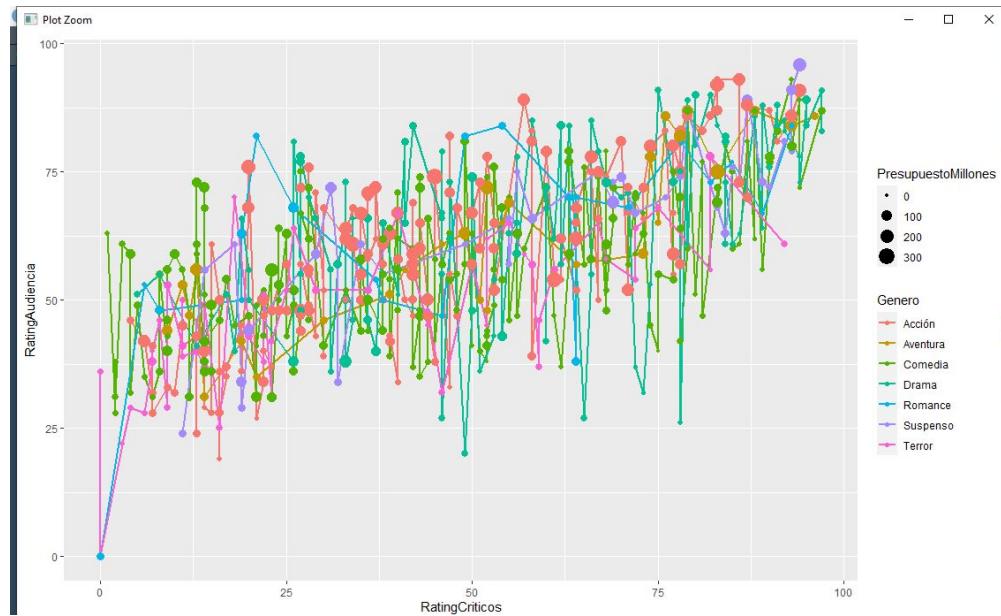
#Una solución a esto es por medio de **xlab()**

```
B + geom_point(aes(x= PresupuestoMillones)) + xlab("PresupuestoMillones")
```



#Volviendo al ejemplo de las líneas, podemos modificar su tamaño para que se puedan ver los puntos fácilmente

```
B + geom_line(size= 1) + geom_point()
```



## Mapeando y Estableciendo

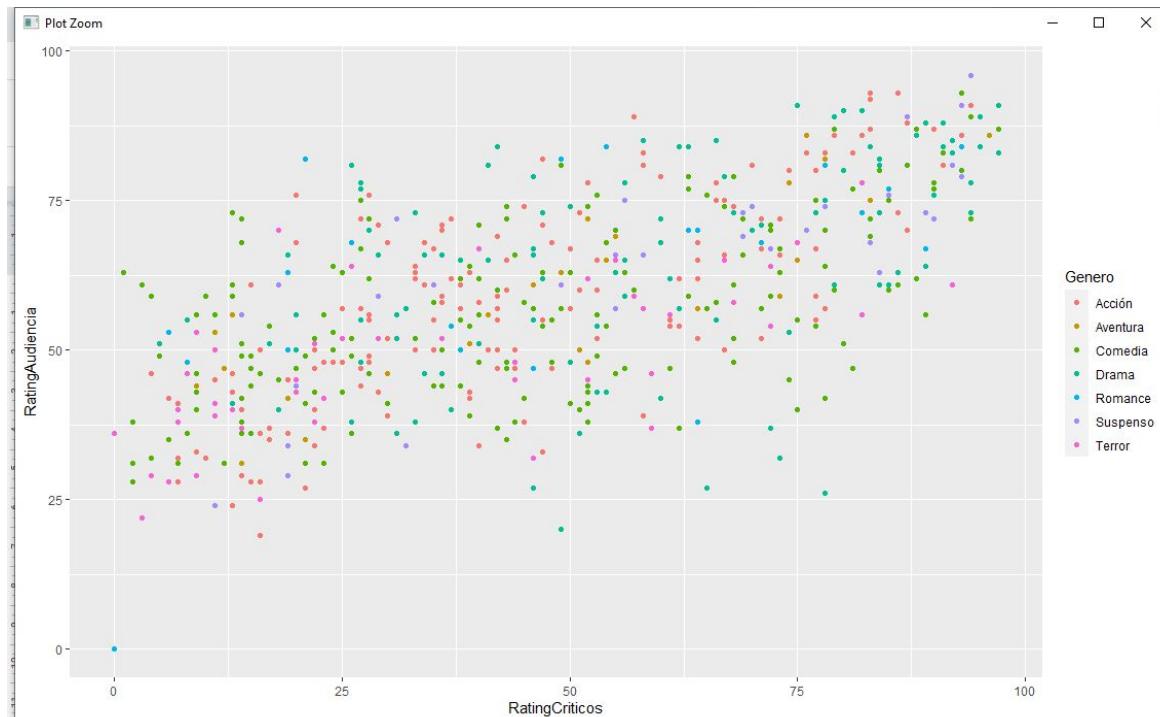
#Mapear contra Establecer

```
C <- ggplot(data= datos.peliculas, aes(x=RatingCriticos, y=RatingAudien  
C + geom_point()
```

#Agregado de color por mapeo y estableciendo

#Mapeo qua involucra el cambio de algo en función de una variable y depende de aes()

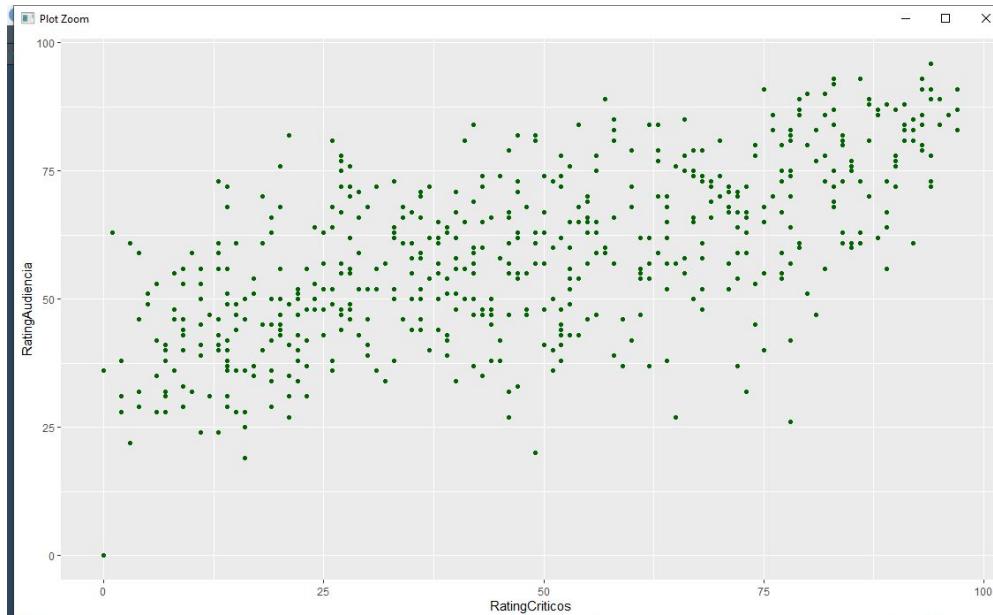
```
C <- ggplot(data= datos.peliculas, aes(x=RatingCriticos, y=RatingAudien  
C + geom_point(aes(color=Genero)) #gráfico es de este  
C + geom_point(aes(size= PresupuestoMillones))
```



#Estableciendo, es directo y no depende de la variable

```
C + geom_point(color="Darkgreen") #gráfico es de este  
C + geom_point(color= "Blue")
```

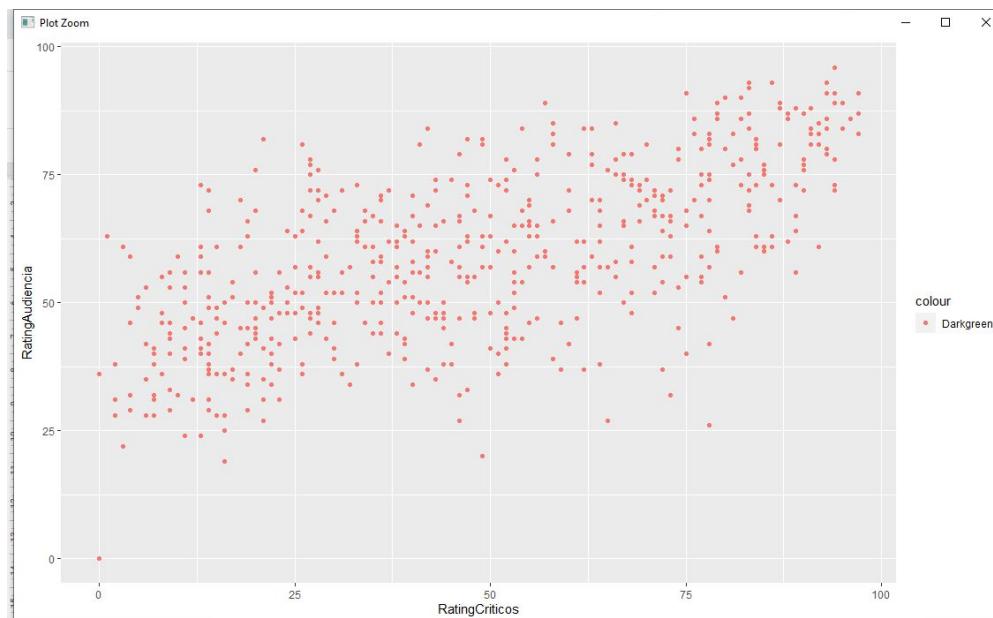
```
C + geom_point(size= 4)
```



```
#ERROR
```

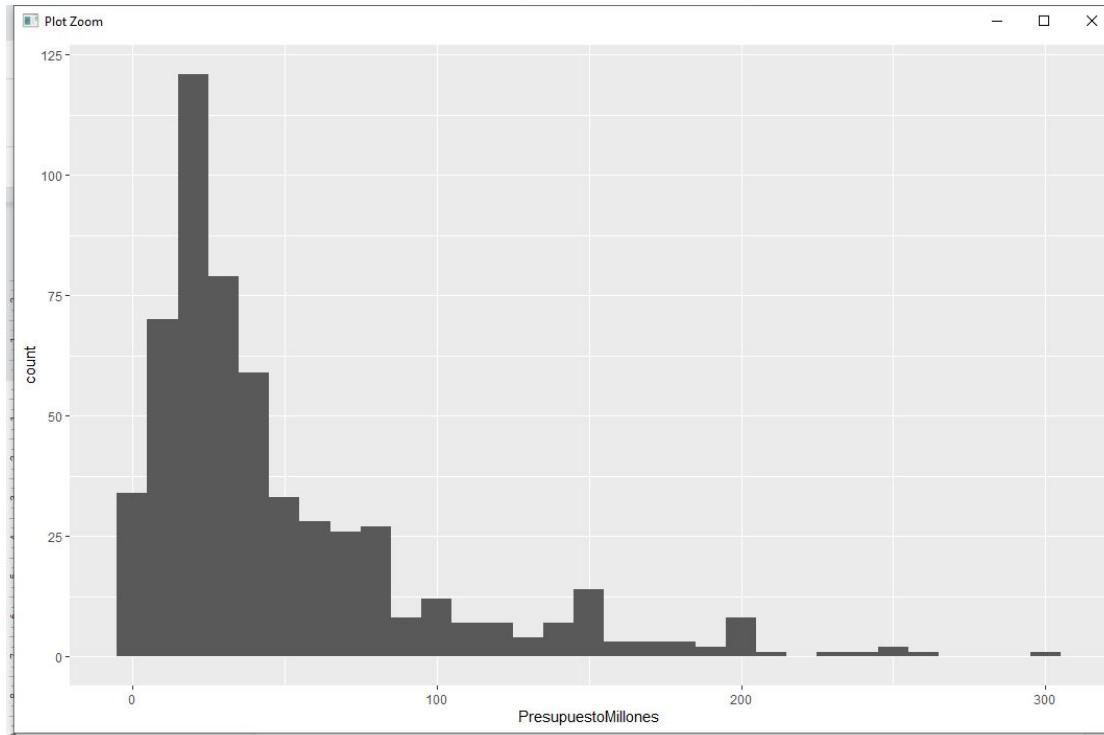
```
C + geom_point(aes(color="Darkgreen")) #gráfico es de este
```

```
C + geom_point(aes(size= 20))
```

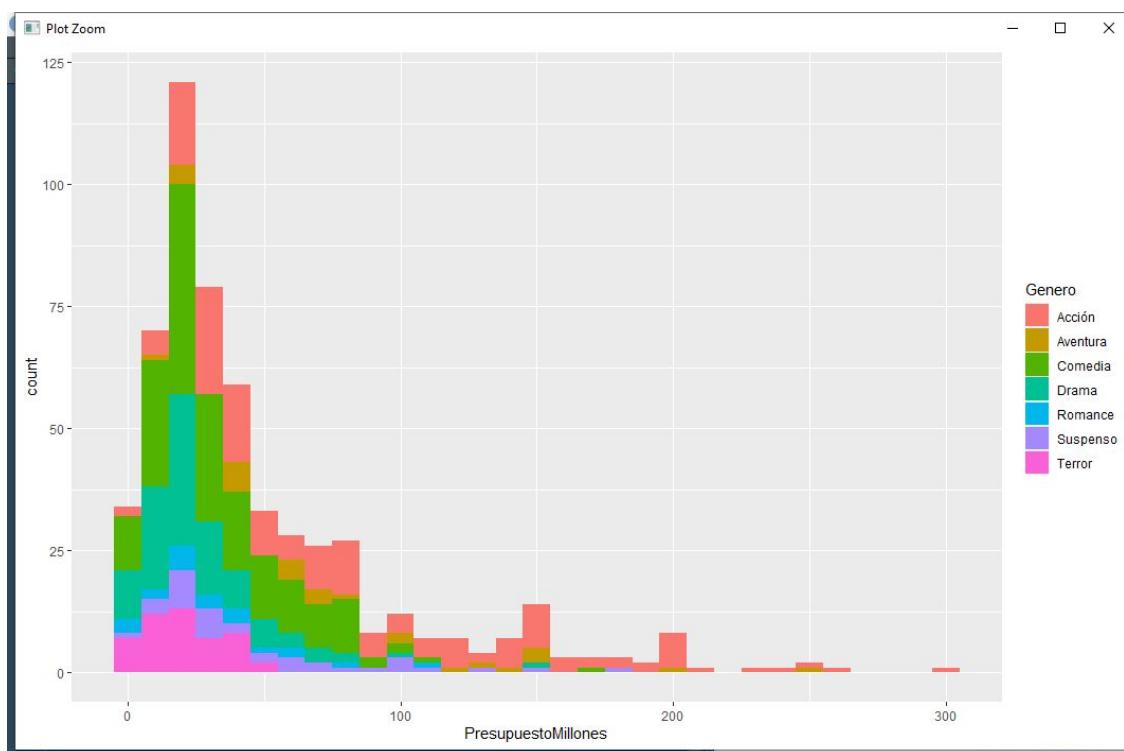
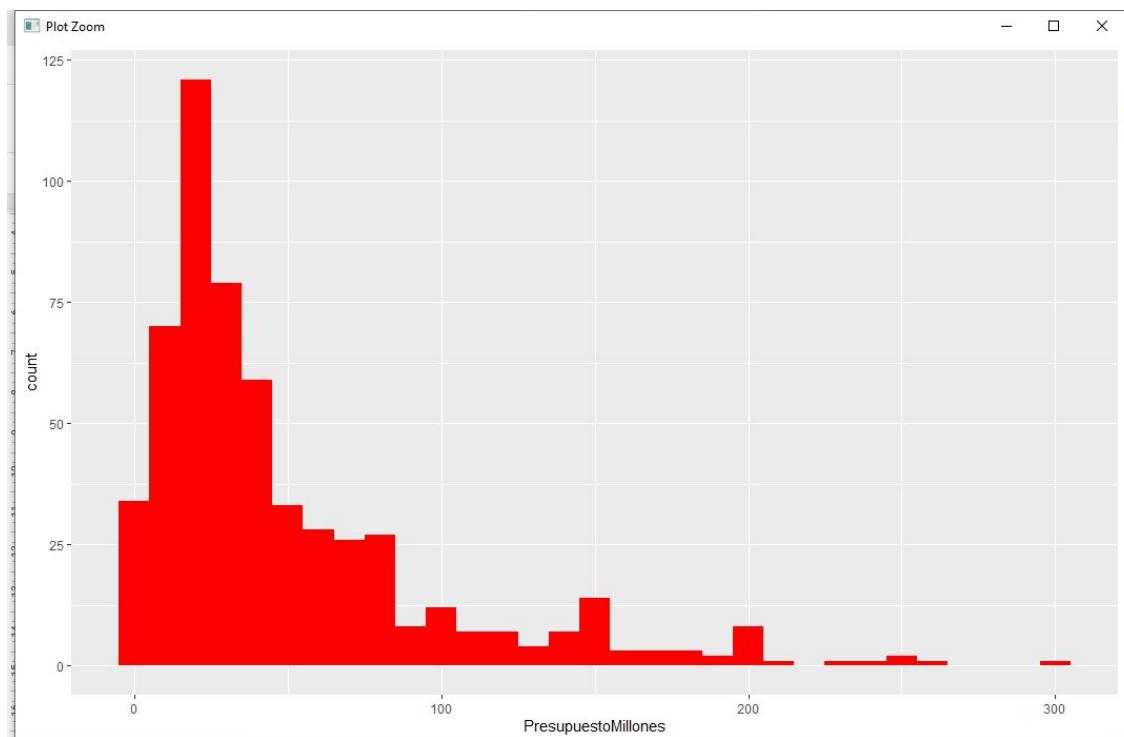


## Histogramas y diagramas de densidad

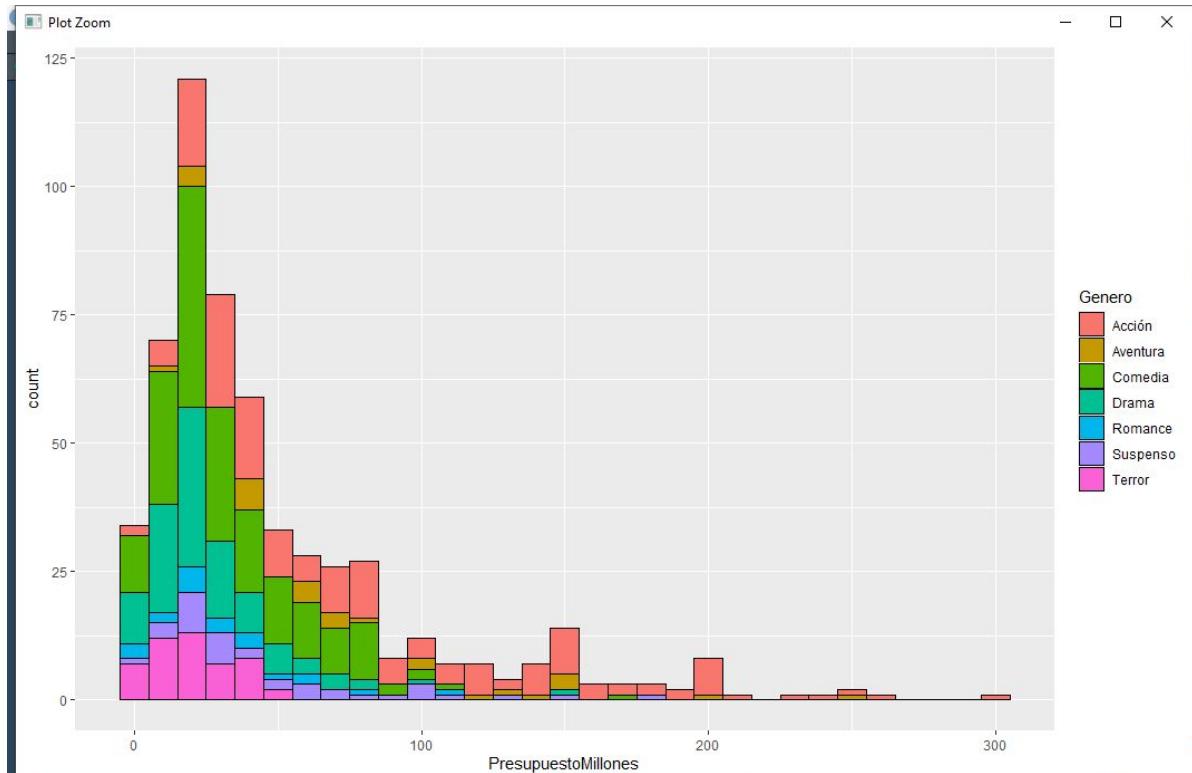
```
#Histogramas y diagramas de densidad  
D <- ggplot(data=datos.peliculas, aes(x=PresupuestoMillones))  
D + geom_histogram(binwidth = 10)  
#binwidth da el ancho del histograma
```



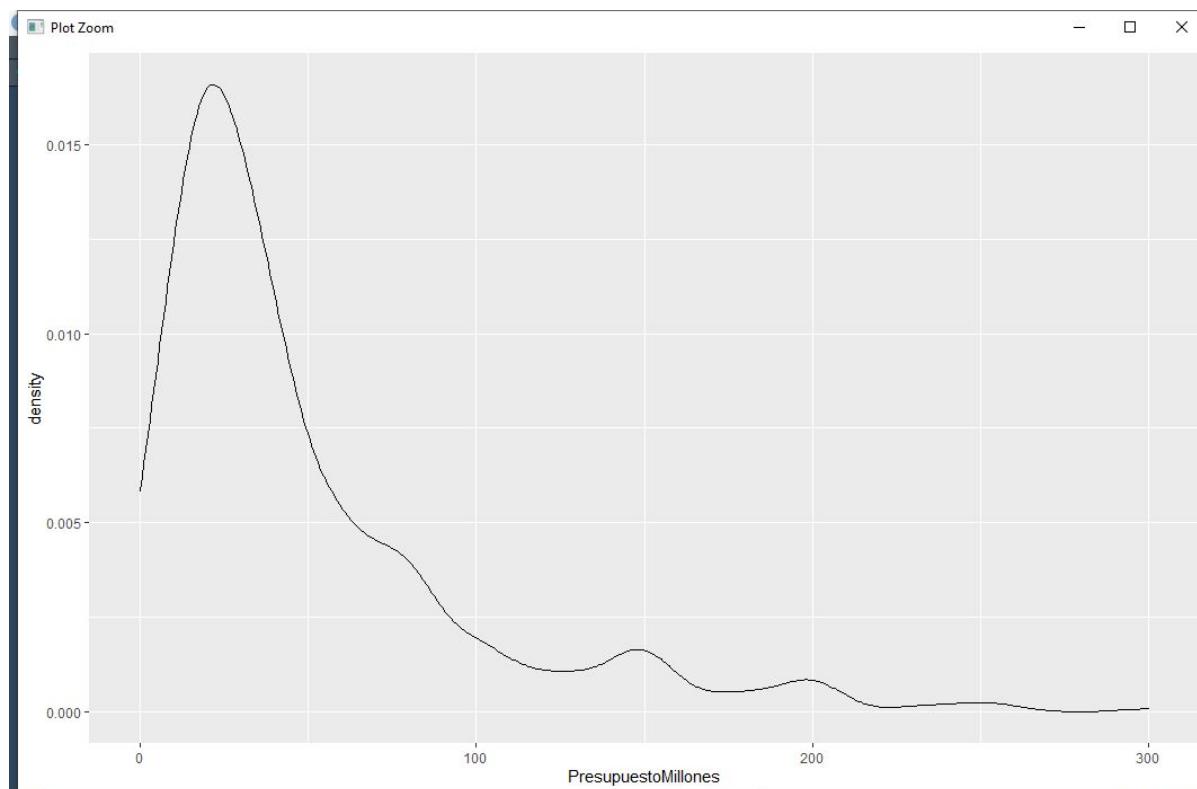
```
#Aregar color  
D + geom_histogram(binwidth = 10, fill ="red") #Establecer  
D + geom_histogram(binwidth = 10, aes(fill = Genero)) #Mapear
```



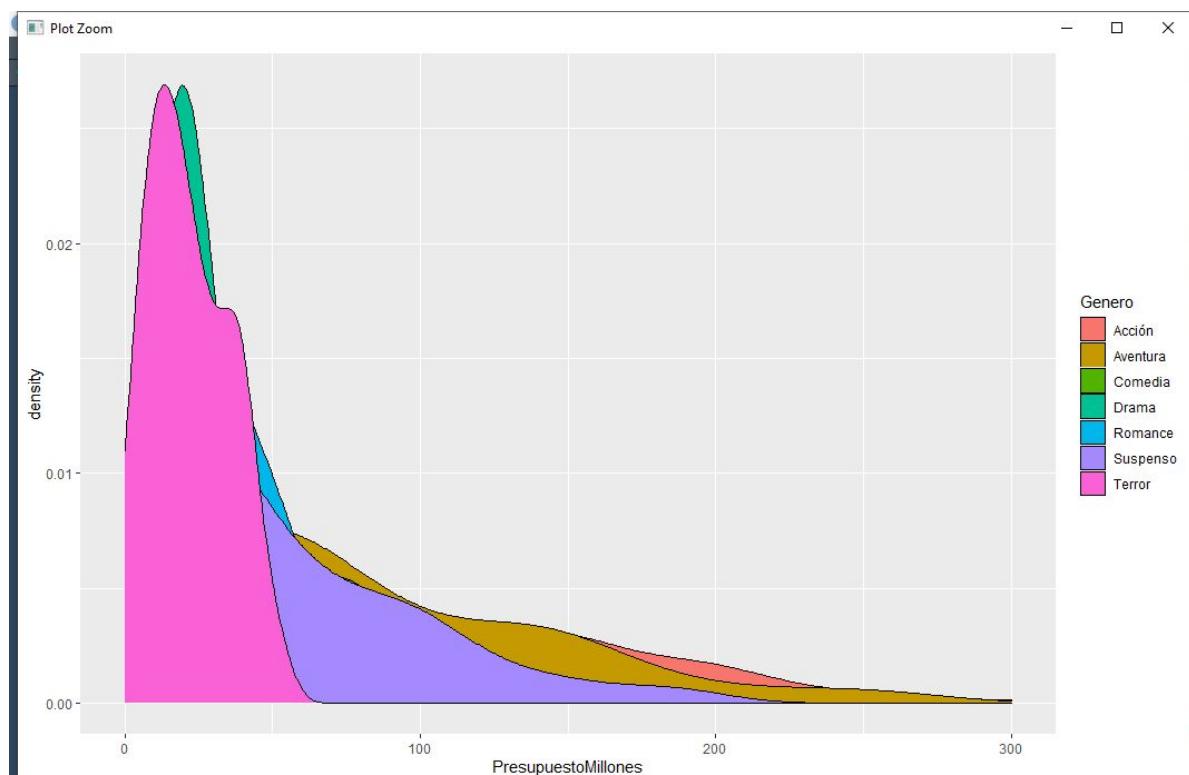
```
#Aregar borde, aquí color da bordes de en lugar del color propiamente  
D + geom_histogram(binwidth = 10, aes(fill = Genero), color = "black")
```



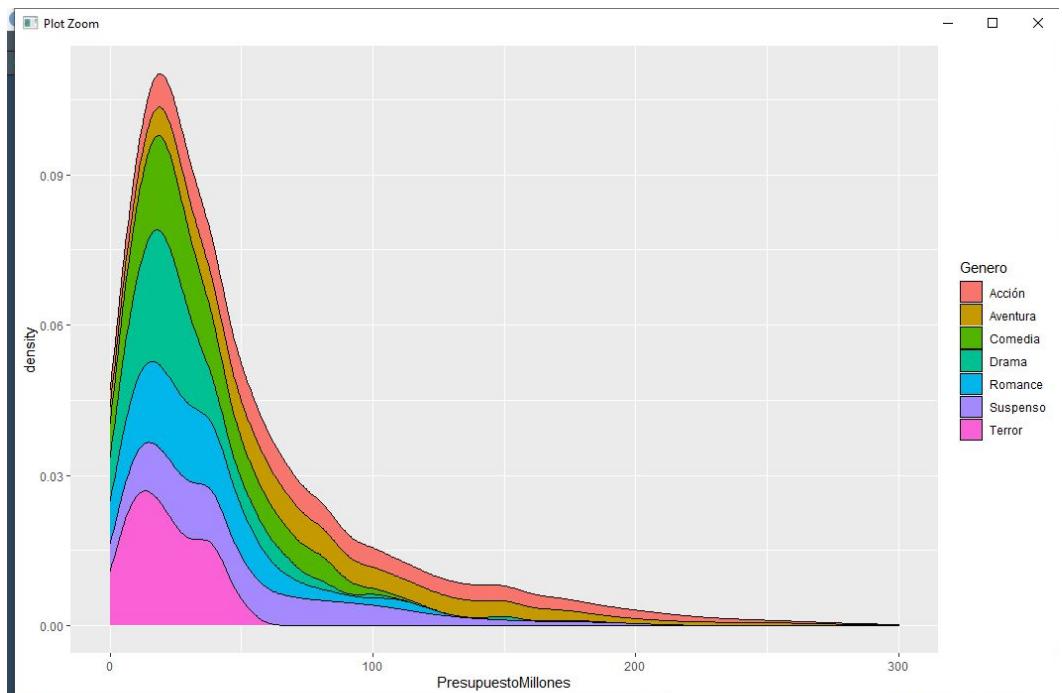
```
#Gráficos de densidad  
D +geom_density()
```



```
D +geom_density(aes(fill= Genero))
```



```
D +geom_density(aes(fill= Genero), position = "stack")
```

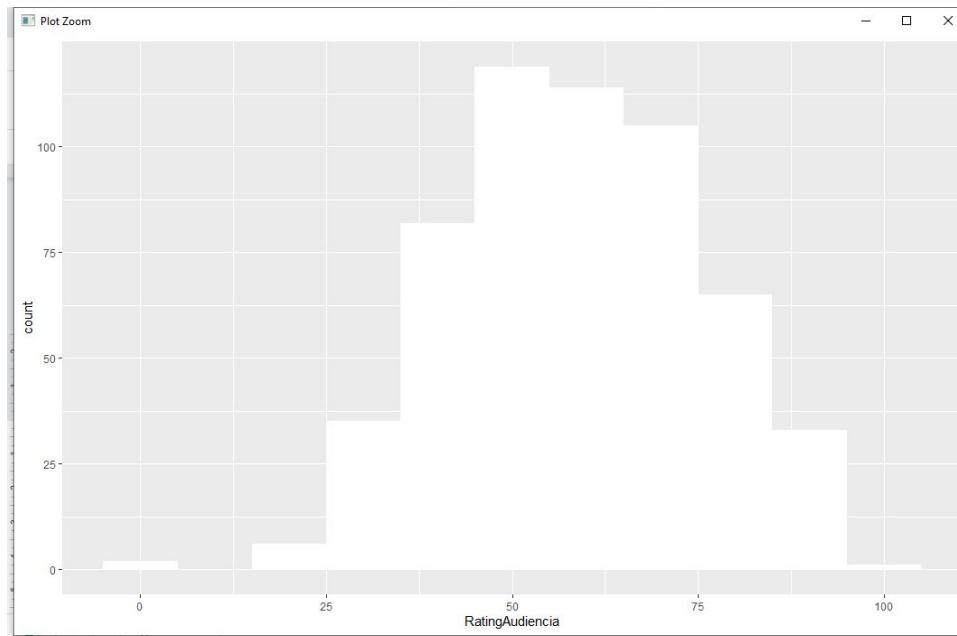


## Tips para capa inicial

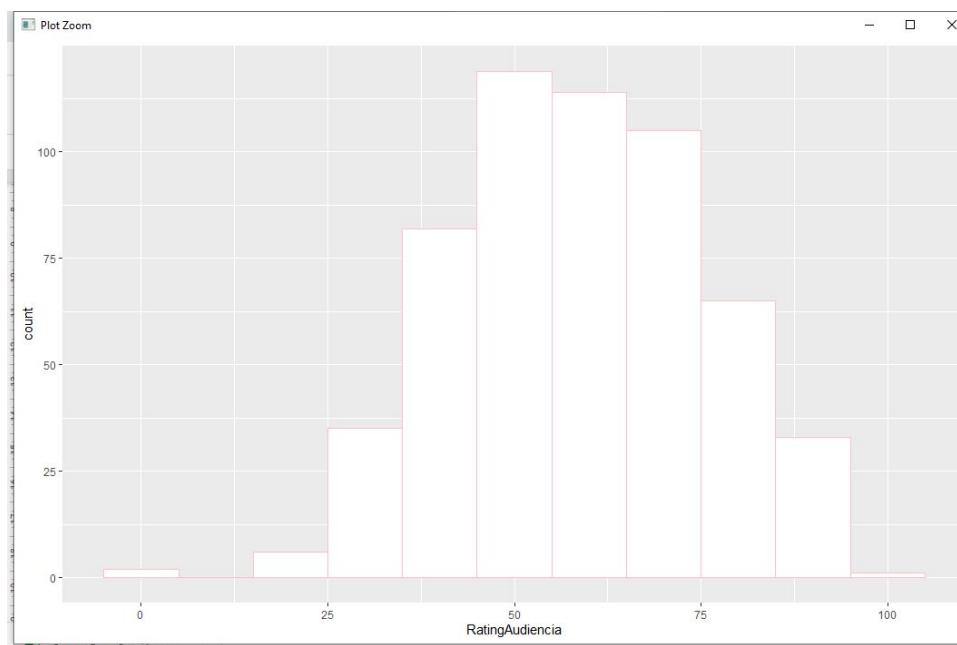
#Tips de capa inicial

```
E<- ggplot(data=datos.peliculas, aes(x=RatingAudiencia))
```

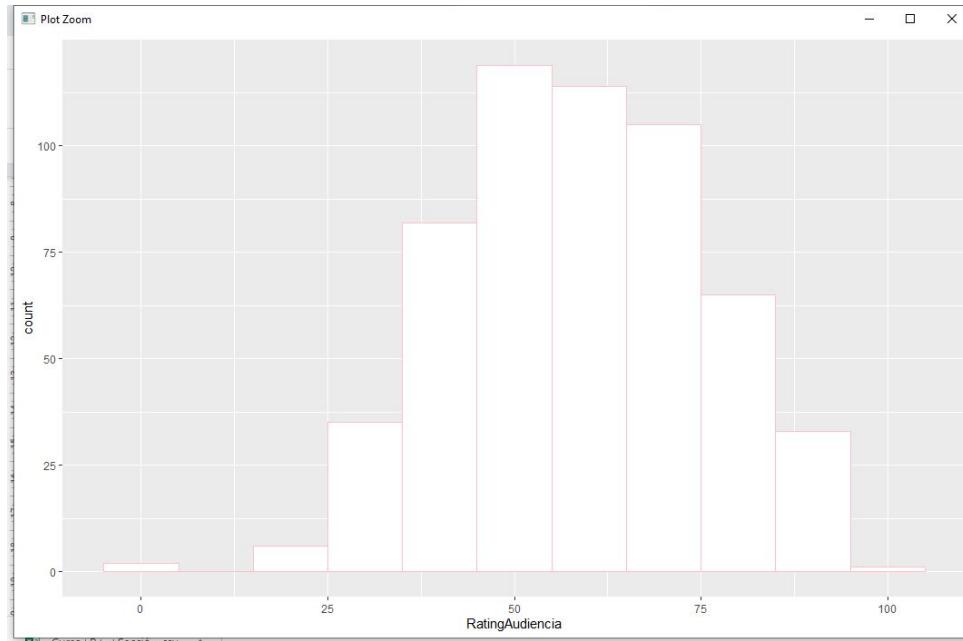
```
E + geom_histogram(binwidth = 10, fill="White")
```



```
E + geom_histogram(binwidth = 10, fill="White", color="pink")
```

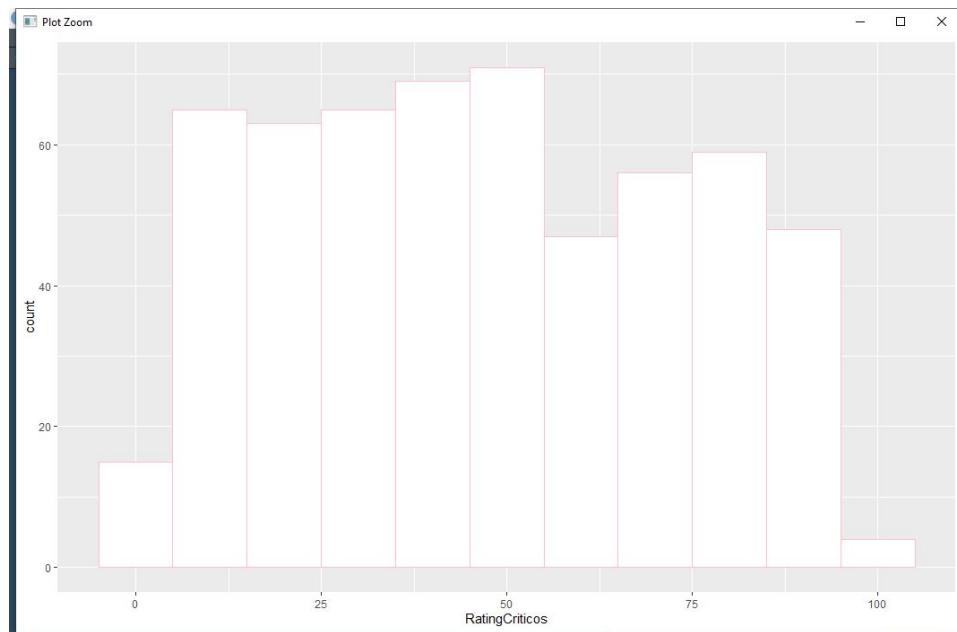


```
#Otra forma de hacer lo de arriba  
E <- ggplot(data=datos.peliculas)  
E + geom_histogram(aes(x=RatingAudencia),  
  binwidth = 10, fill="White", color="pink")
```



```
#Este último es el que tiene mayor flexibilidad, es adecuado cuando vamos a necesitar hacer  
muchas y varias cosas diferentes con el marco de datos, o cuando no estamos muy seguros  
qué hacer con él.
```

```
#Lo mismo pero ahora con los críticos  
E + geom_histogram(aes(x=RatingCriticos),  
  binwidth = 10, fill="White", color="pink")
```



#Una última opción es dejar el esqueleto de ggplot sin nada

```
E <- ggplot()
```

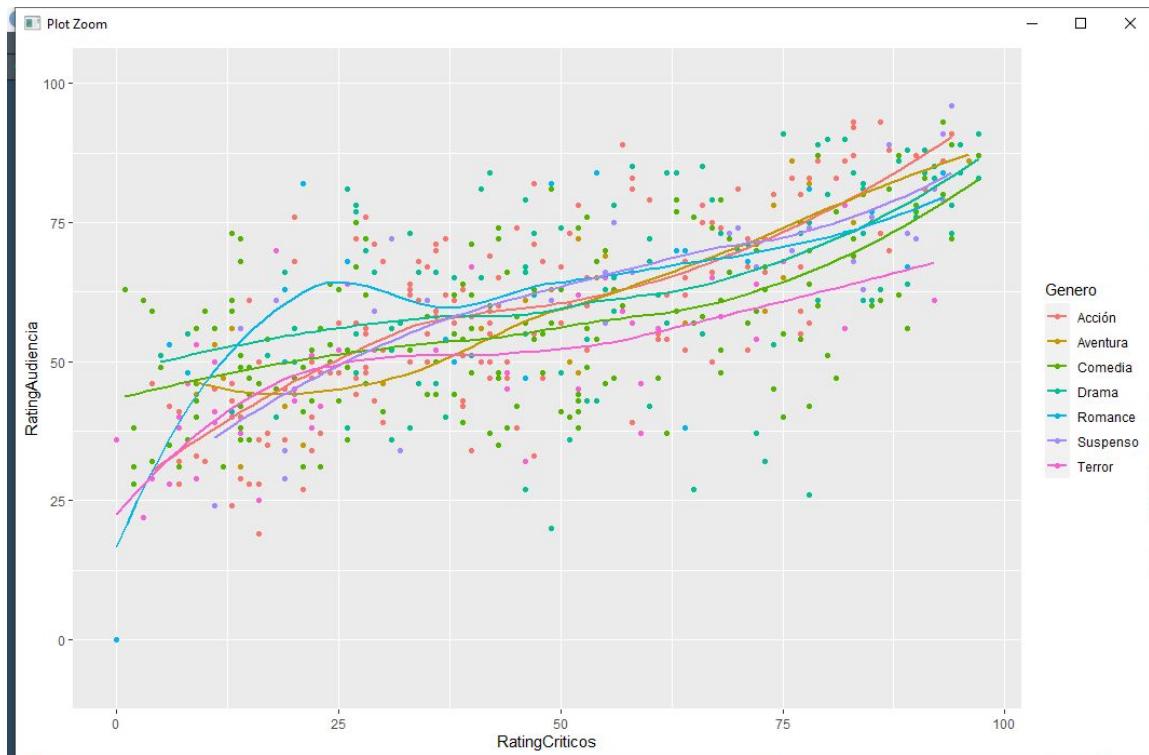
## Transformaciones Estadísticas

#Transformaciones estadísticas

### #Geom\_smooth()

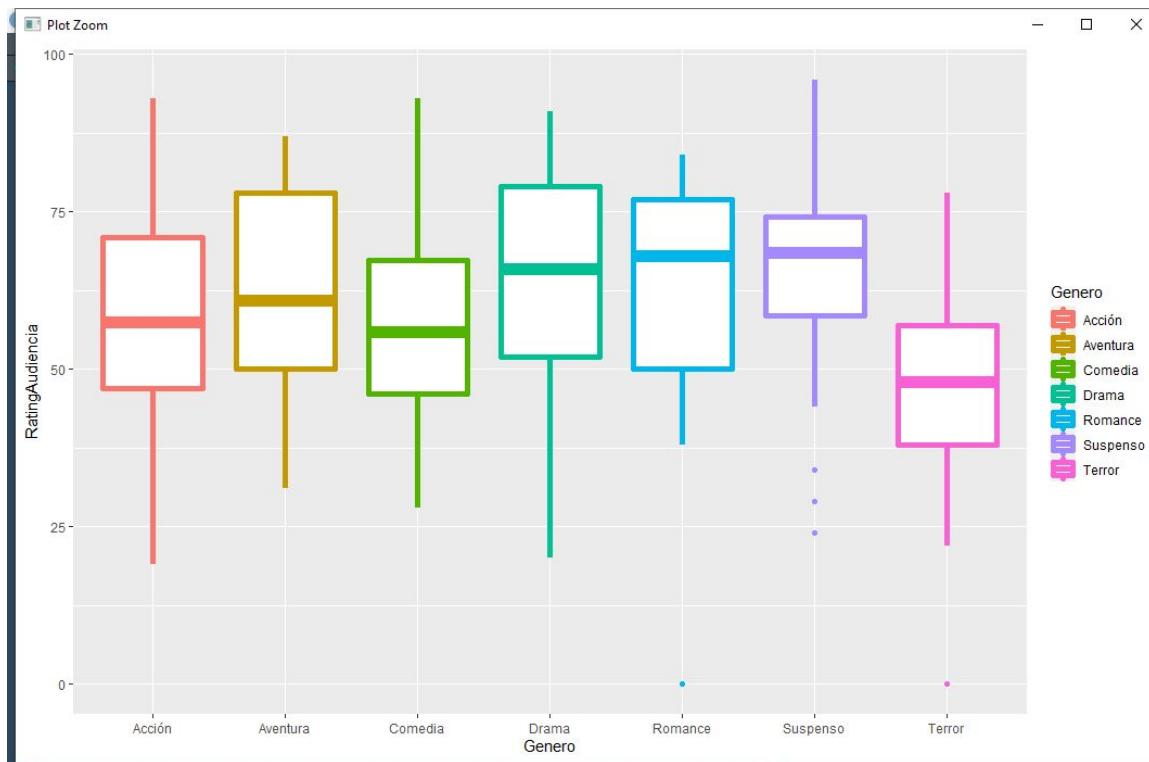
#Mediante estas líneas geom\_smooth permite observar patrones que serían complicados de ver cuando los datos están dispersos

```
G <- ggplot(data= datos.peliculas, aes(x= RatingCriticos, y= RatingAudien  
cia,  
color= Genero))  
G + geom_point() + geom_smooth(fill=NA)
```



### #Geom\_boxplot

```
G2 <- ggplot(data=datos.peliculas, aes(x= Genero, y= RatingCriticos,  
color= Genero))
```



#Otras opciones de geom\_boxpot

G2 + geom\_boxplot(size= 2) + geom\_point()

G2 + geom\_boxplot(size=2) + geom\_jitter()



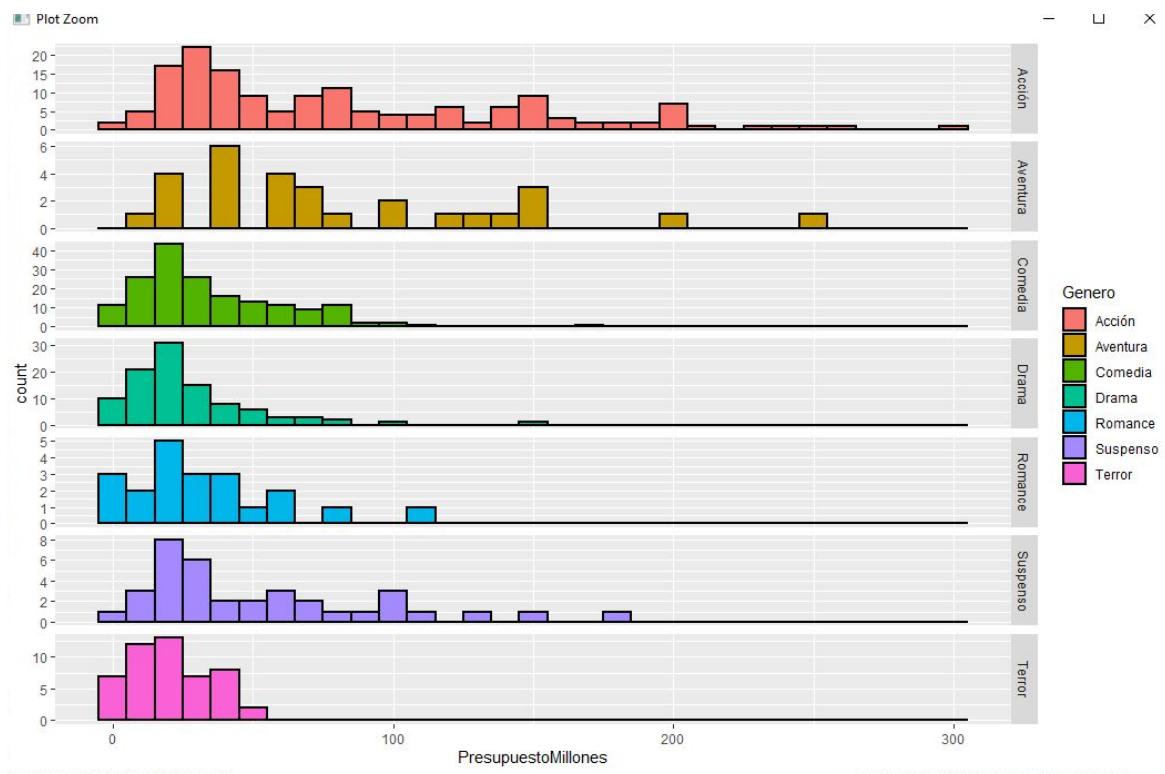
## Uso de Facetas

#Usando Facetas

```
H <- ggplot(data=datos.peliculas, aes(x= PresupuestoMillones))  
H + geom_histogram(binwidth =10, aes(fill=Genero), color= "Black", size= 1)
```

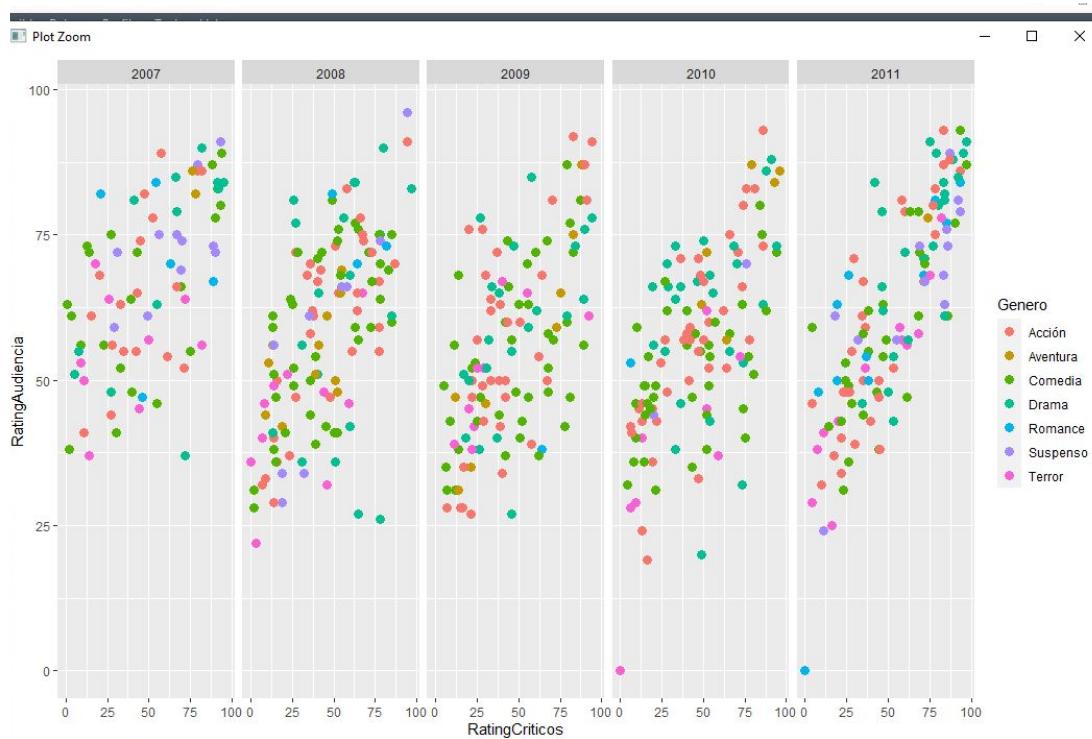
#Facetas, por medio de éstas podemos tener un histograma, o cualquier geometría, para cada uno de los factores de un conjunto de datos. En nuestro caso tenemos uno para cada uno de los géneros

```
H + geom_histogram(binwidth =10, aes(fill=Genero), color= "Black", size= 1) +  
facet_grid(Genero~.,scales = "free")
```



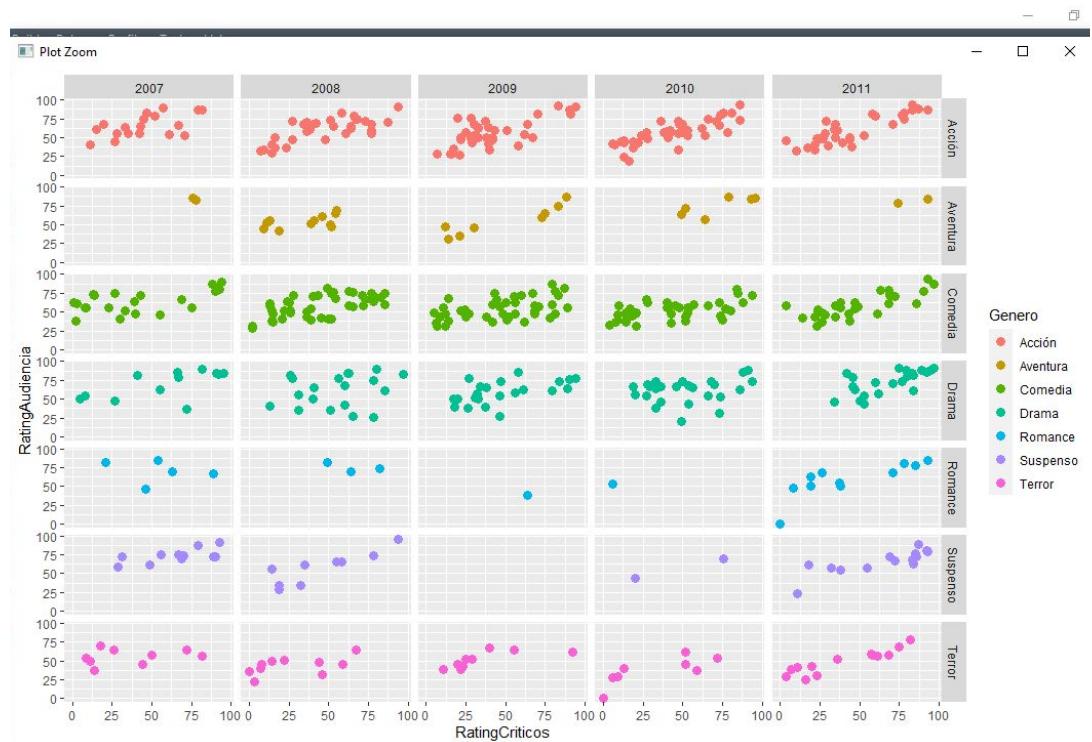
#Facetas con gráficos de dispersión

```
I <- ggplot(data=datos.peliculas, aes(x=RatingCriticos, y=RatingAudienca,  
color=Genero))  
I + geom_point(size= 3) + facet_grid(.~Año) #Columnas
```

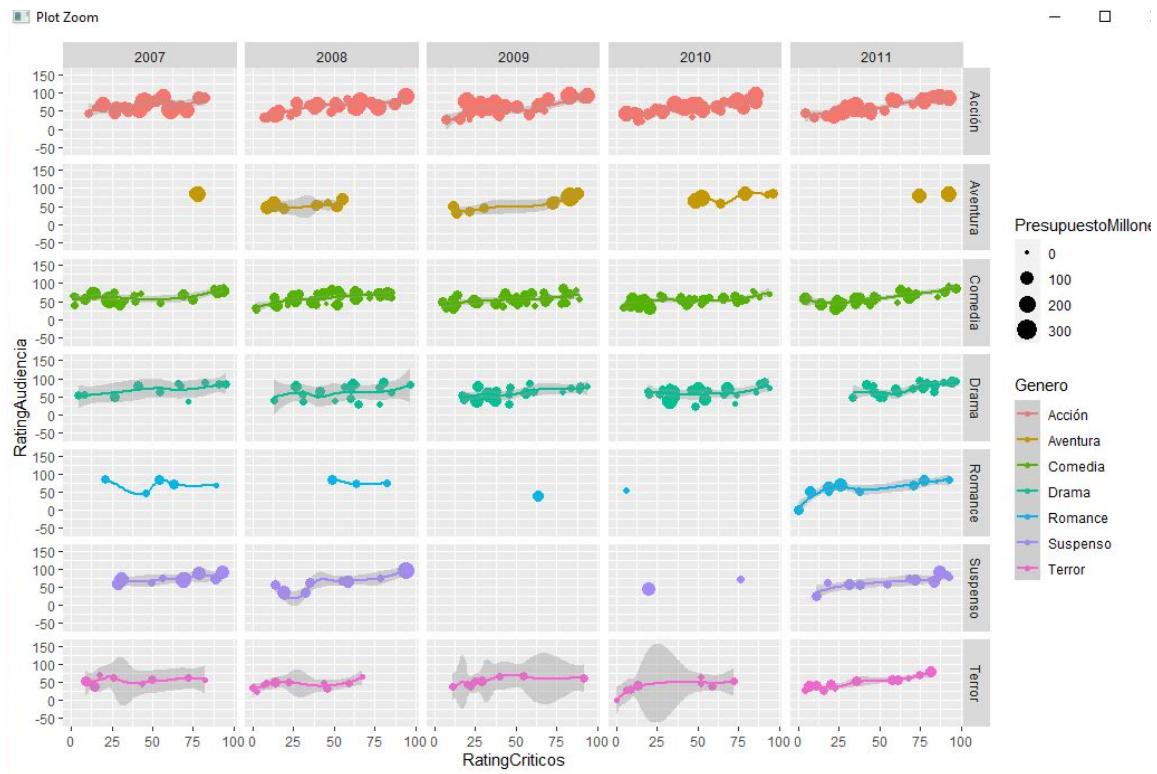


I + geom\_point(size= 3) + facet\_grid(Año~.) #Filas

I + geom\_point(size= 3) + facet\_grid(Genro~Año) #Genro en filas y columnas de Año



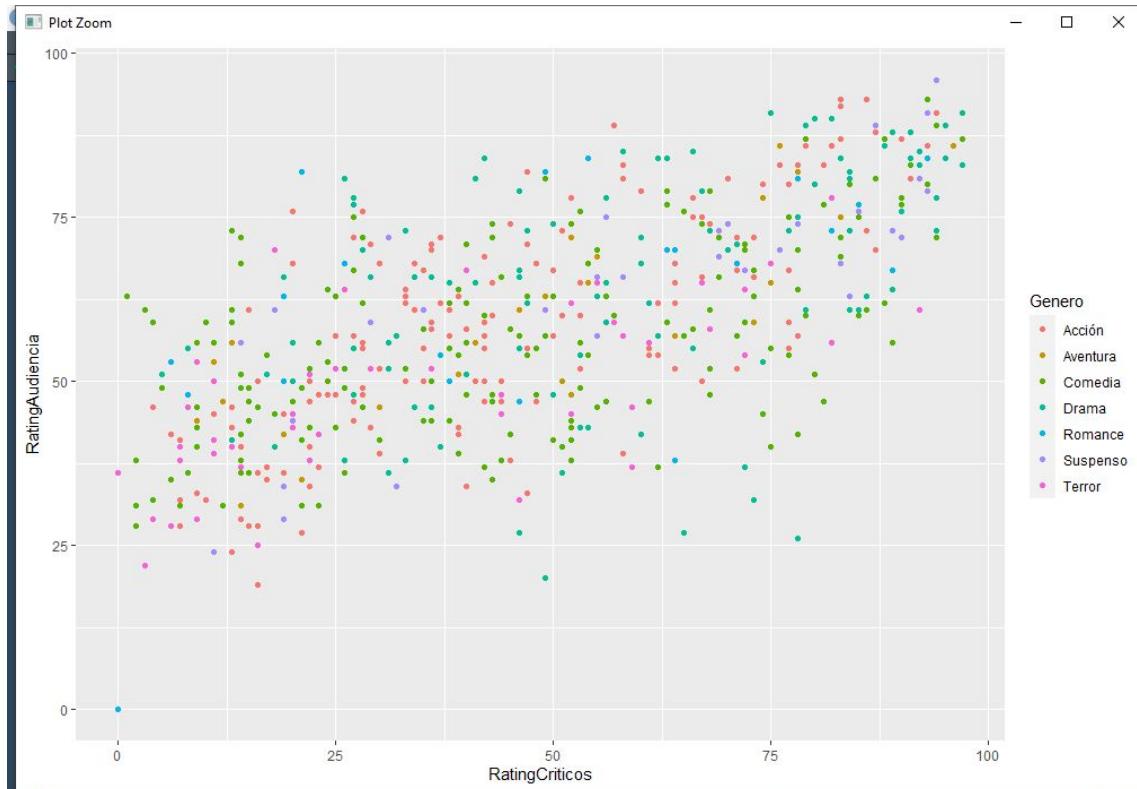
```
I + geom_point(aes(size= PresupuestoMillones)) + facet_grid(Genero~Año) +  
geom_smooth()
```



## Coordenadas

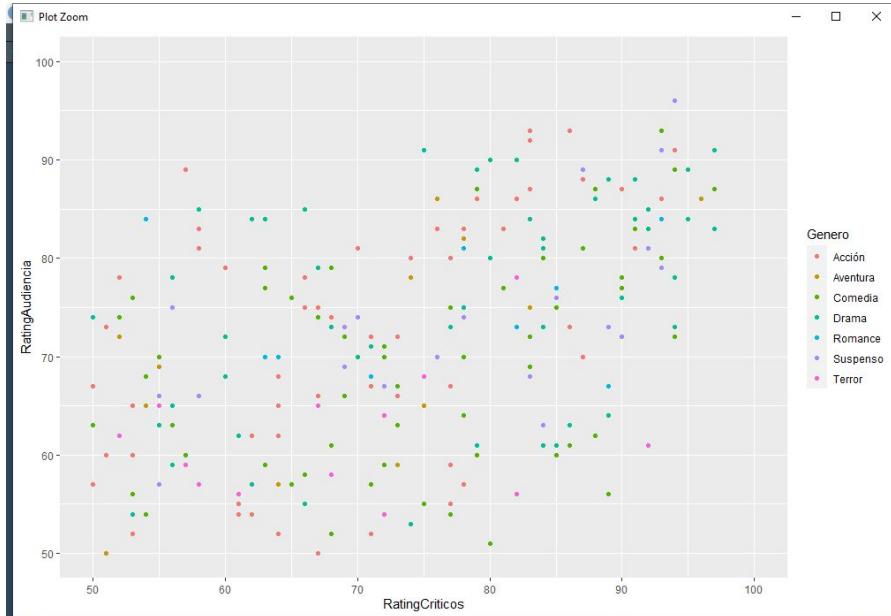
#Coordinandas

```
J <- ggplot(data=datos.peliculas, aes(x=RatingCriticos, y=RatingAudiencia,  
color=Genero))  
J + geom_point()
```



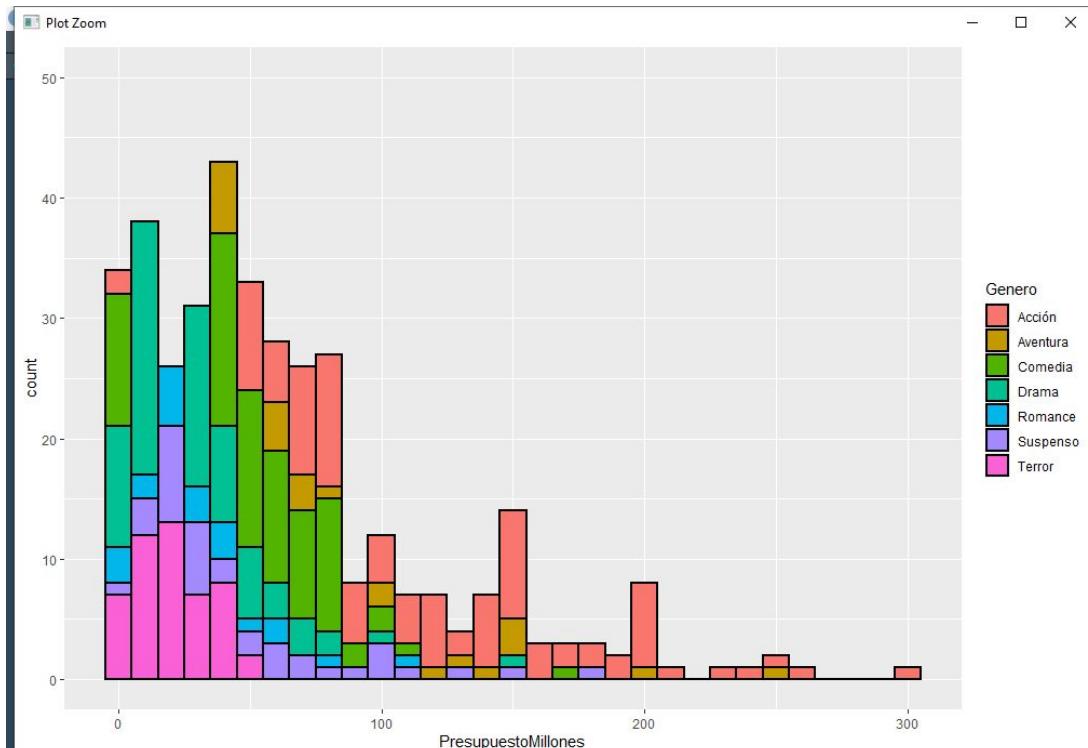
#Si + xlim() y + ylim() podemos lograr seleccionar una sección de todo el sistema de coordenadas

```
J + geom_point() + xlim(50, 100) + ylim(50, 100)
```

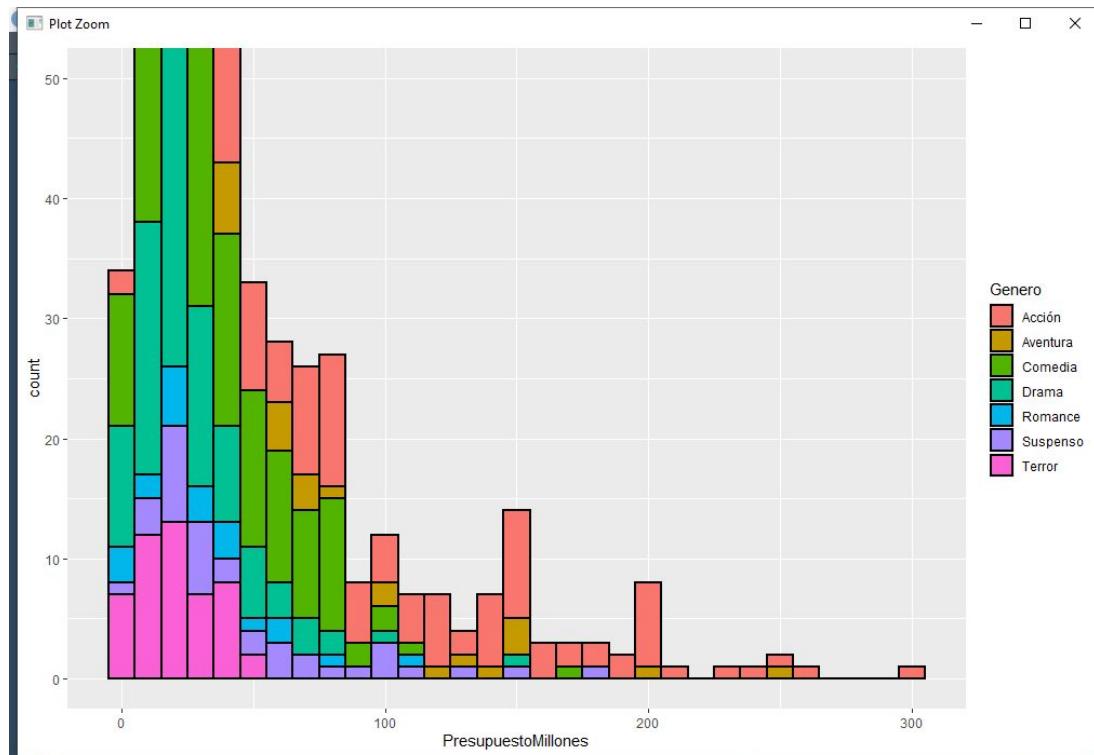


#Sin embargo no siempre es útil, como en el caso de un histograma

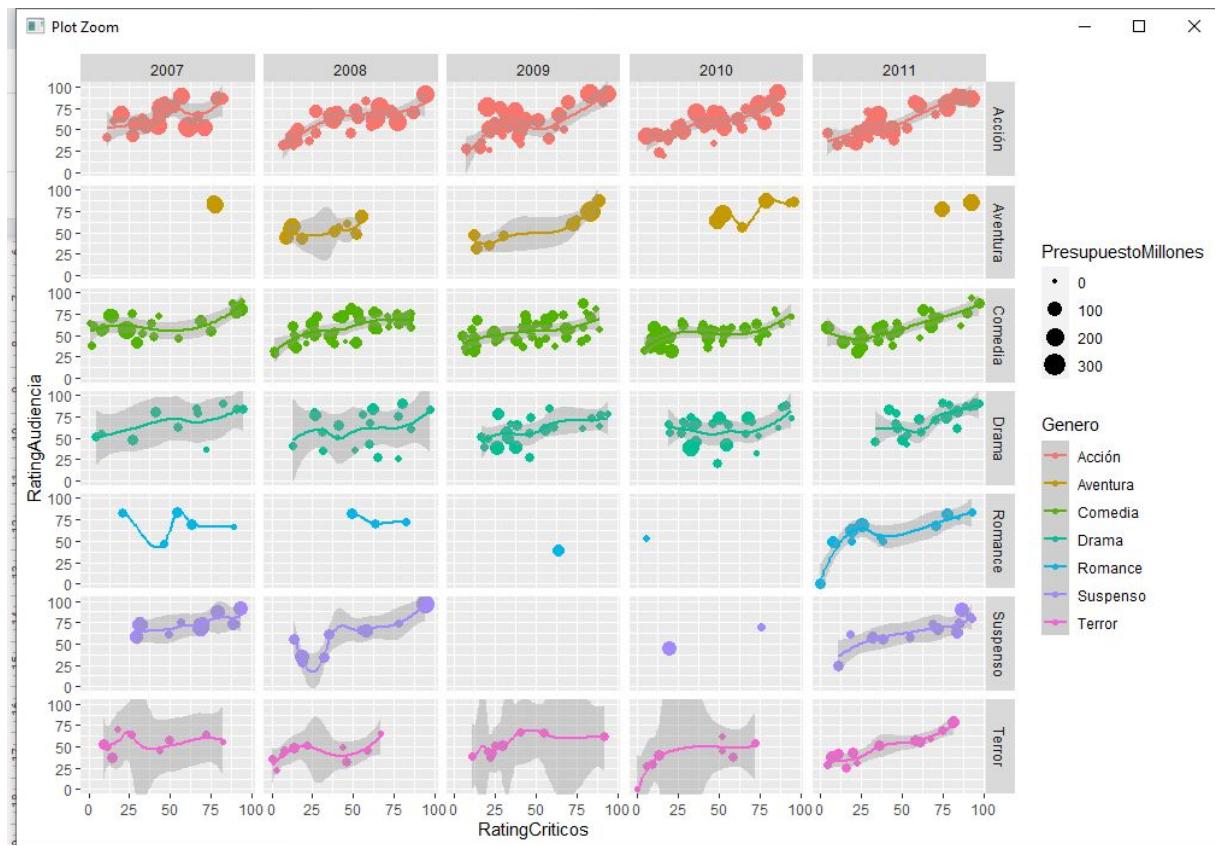
```
K <- ggplot(data=datos.peliculas, aes(x=PresupuestoMillones))
K + geom_histogram(binwidth=10, aes(fill=Genero), color="black", size= 1) +
  ylim(0,50)
```



```
#El problema con esto anterior es que se cortan los datos y se pierde mucha información
#Para resolver esto podemos usar coord_cartesian() que también hace uso de xlim() y ylim()
K + geom_histogram(binwidth=10, aes(fill=Genero), color="black", size= 1) +
coord_cartesian(ylim=c(0,50))
```

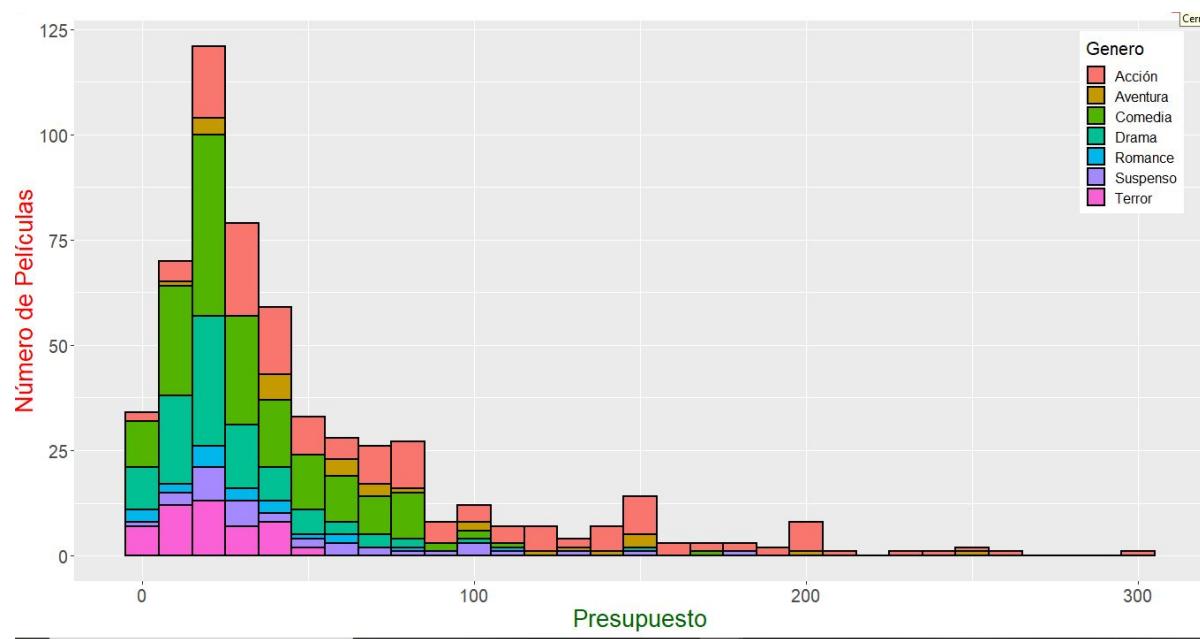


```
L <- ggplot(data=datos.peliculas, aes(x=RatingCriticos, y=RatingAudiencia, color=Genero))
L + geom_point(aes(size=PresupuestoMillones)) + geom_smooth() +
facet_grid(Genero~Año) + coord_cartesian(ylim=c(0,100))
```



## Formato

```
#Formato  
N + xlab("Presupuesto") + ylab("Número de Películas") +  
  theme(axis.title.x = element_text(color = "Darkgreen", size=20),  
        axis.title.y = element_text(color= "Red", size=20),  
        axis.text = element_text(size=15), #si no pongo .y o .x se modifican ambos  
        legend.title=element_text(size=15),  
        legend.text = element_text(size= 12),  
        legend.position = c(0.98,0.98), #Determina la posición en el gráfico  
        legend.justification = c(1,1)) #Ancla un punto de referencia para que no se salga del  
margen
```



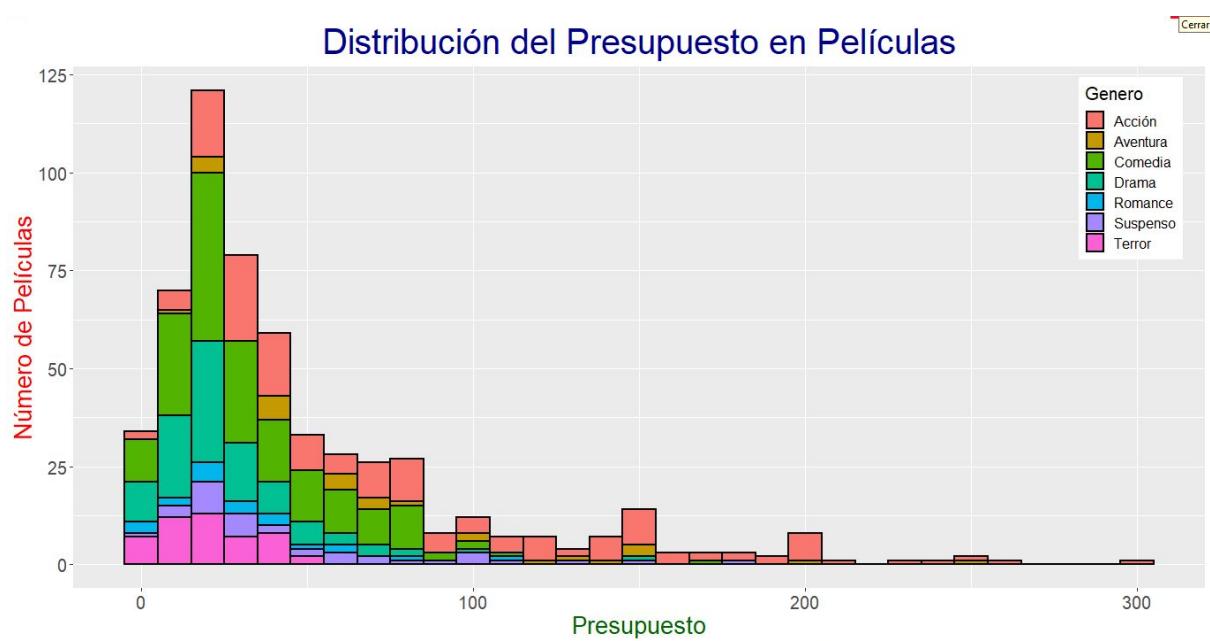
```
#Título del diagrama
```

```
N + xlab("Presupuesto") + ylab("Número de Películas") +  
  ggtitle("Distribución del Presupuesto en Películas") +  
  theme(axis.title.x = element_text(color = "Darkgreen", size=20),  
        axis.title.y = element_text(color= "Red", size=20),  
        axis.text = element_text(size=15), #si no pongo .y o .x se modifican ambos
```

```

legend.title=element_text(size=15),
legend.text = element_text(size= 12),
legend.position = c(0.98,0.98), #Determina la posición en el gráfico
legend.justification = c(1,1), #Ancla un punto de referencia para que no se salga del
margen
plot.title = element_text(color = "darkblue", size = 30, hjust = .5)
)

```

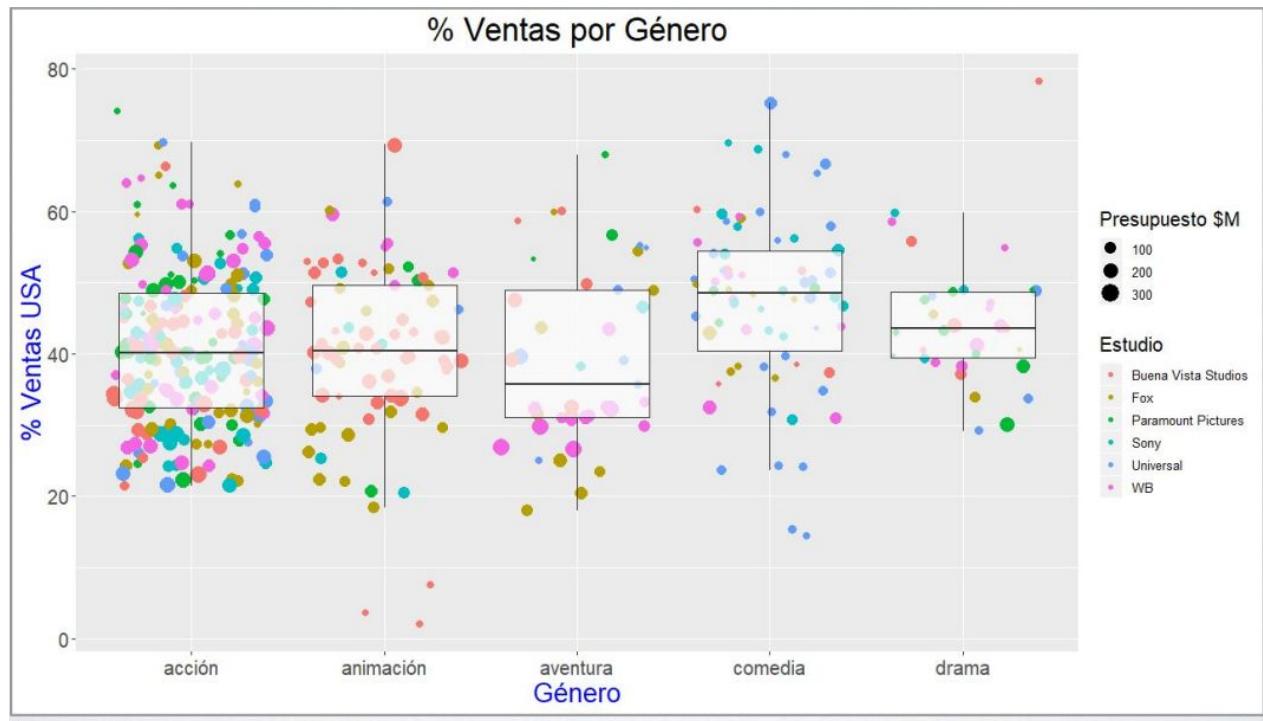


## Práctica Final

La página de reseñas de películas quedó muy satisfecha con tu entregable pasado y ahora tiene un nuevo requisito para ti.

El consultor pasado creó un gráfico para ellos, el cual puedes ver en la siguiente diapositiva. Sin embargo, el código utilizado para crear el diagrama se ha perdido y no lo pueden recuperar. Tu tarea es volver a escribir el código para crear el gráfico lo más parecido que se pueda al original.

Te han dado un set de datos nuevo.



```
#Importa los datos
peliculas <- read.csv(file.choose())

#Análisis Exploratorio

head(peliculas) #filas superiores
summary(peliculas) #resumen de las columnas
str(peliculas) #estructura del set de datos

#Activar GGPlot2
#Usar install.package("ggplot2") en caso de no tener el paquete descargado
library(ggplot2)

#Fuera de alcance pero esta interesante:
ggplot(data=peliculas, aes(x=Día.de.la.Semana..lanzamiento.)) + geom_bar()
#¿Te das cuenta? No ha habido estrenos de películas en un Lunes.

#Ahora vamos a filtrar nuestro set de datos para dejar únicamente
#los Géneros y los Estudios en los que estamos interesados
#Empezaremos con el filtro de Género. Usaremos el operador lógico
#"or" para seleccionar múltiples Géneros:
filtro1 <- (peliculas$Género == "acción") | (peliculas$Género == "aventura") |
(peliculas$Género == "animación") | (peliculas$Género == "comedia") | (peliculas$Género
== "drama")

#Ahora hagamos lo mismo para los Estudios:
filtro2 <- (peliculas$Estudio == "Buena Vista Studios") | (peliculas$Estudio == "WB") |
(peliculas$Estudio == "Fox") | (peliculas$Estudio == "Universal") | (peliculas$Estudio ==
"Sony") | (peliculas$Estudio == "Paramount Pictures")

#Aplica los filtros de las filas al marco de datos
peliculas2 <- peliculas[filtro1 & filtro2,]
```

```
#Prepara los datos del gráfico y las capas de estéticas
#Nota que no le cambiamos el nombre a las columnas
#Usa str() o summary() para encontrar el nombre correcto de las columnas
str(peliculas2)
p <- ggplot(data=peliculas2, aes(x=Género, y=Venta...USA))
p #No pasa nada porque se necesita una geometría

#Agrega una capa con geometría de puntos
p + geom_point()

#Puedes agregar un boxplot en lugar de los puntos
p + geom_boxplot()

#Nota que los valores atípicos son parte de la capa del boxplot
#Usaremos esa observación después (*)

#Agrega los puntos
p + geom_boxplot() + geom_point()
#No es exactamente lo que estábamos buscando

#Cambia los puntos por el jitter
p + geom_boxplot() + geom_jitter()

#Posiciona el boxplot por encima del jitter
p + geom_jitter() + geom_boxplot()

#Agrega transparencia al boxplot
p + geom_jitter() + geom_boxplot(alpha=0.7)

#Ahora puedes agregar tamaño y color a los puntos:
p + geom_jitter(aes(size=Presupuesto...mill., color=Estudio)) +
  geom_boxplot(alpha=0.7)
#¿Puedes ver los puntos negros que aún están visibles?
#¿De dónde vienen?
```

```
#Son parte del boxplot - ¿Recuerdas la observación (*) que hicimos arriba?
```

```
#Vamos a quitarlos:
```

```
p + geom_jitter(aes(size=Presupuesto...mill., color=Estudio)) +  
  geom_boxplot(alpha = 0.7, outlier.colour = NA)
```

```
#Almacenamos nuestro progreso en un nuevo objeto:
```

```
q <- p + geom_jitter(aes(size=Presupuesto...mill., color=Estudio)) +  
  geom_boxplot(alpha = 0.7, outlier.colour = NA)  
q
```

```
#Elementos que no son datos (non-data ink)
```

```
q <- q +  
  xlab("Género") #título del eje x  
  ylab("% Ventas USA") #título del eje y  
  ggtitle("Domestic Gross % by Genre", ) #título del diagrama  
q
```

```
#Para lo siguiente se requiere theme()
```

```
#Tema
```

```
q <- q +  
  theme(
```

```
#Título de los ejes:
```

```
  axis.title.x = element_text(color="Blue", size=20),  
  axis.title.y = element_text(color="Blue", size=20),
```

```
#Texto de los ejes:
```

```
  axis.text.x = element_text(size=15),  
  axis.text.y = element_text(size=15),
```

```
#Título del gráfico:
```

```
  plot.title = element_text(color="Black",  
                            size=25,
```

```
hjust = 0.5),
```

```
#Título de la Leyenda:
```

```
legend.title = element_text(size=15),
```

```
#Texto de la Leyenda
```

```
legend.text = element_text(size=10)
```

```
)
```

```
q
```

```
#Toque Final. Esto no lo habíamos visto durante el curso
```

```
#Pero de esta manera puedes cambiar individualmente el título de tu leyenda
```

```
q$labels$size = "Presupuesto $M"
```

```
q
```

### Domestic Gross % by Genre

