R : Data Visualization with ggplot2

Sławek Staworko

University of Lille

2020

Working example

Young people survey

- 1010 participants of Slovakian nationality (aged 15-30)
- 150 questions, some categorical but most on scale of 1 to 5, covering
  - Music and movies preferences
  - Hobbies and interests
  - Phobias, habits, and personality traits
  - demographic information

Obtaining data

Available at Kaggle:
`https://www.kaggle.com/miroslavsabo/young-people-survey`

local copy can be found on the page
`http://researchers.lille.inria.fr/~staworko/r18.html`

we assume survey data is loaded and assigned to a variable
`df ← read.csv('responses.csv', na.strings='')`

The ggplot2 library

- Comprehensive plotting system for R
- Based on a grammar of graphics
- Loaded with `library(ggplot2)`
- installation might be necessary with the command `install.packages("ggplot2")`
- Full documentation on `http://ggplot2.tidyverse.org/index.html`
- Reference index `http://ggplot2.tidyverse.org/reference/index.html`
- Gallery `http://www.ggplot2-exts.org/gallery/`

Grammar of graphics

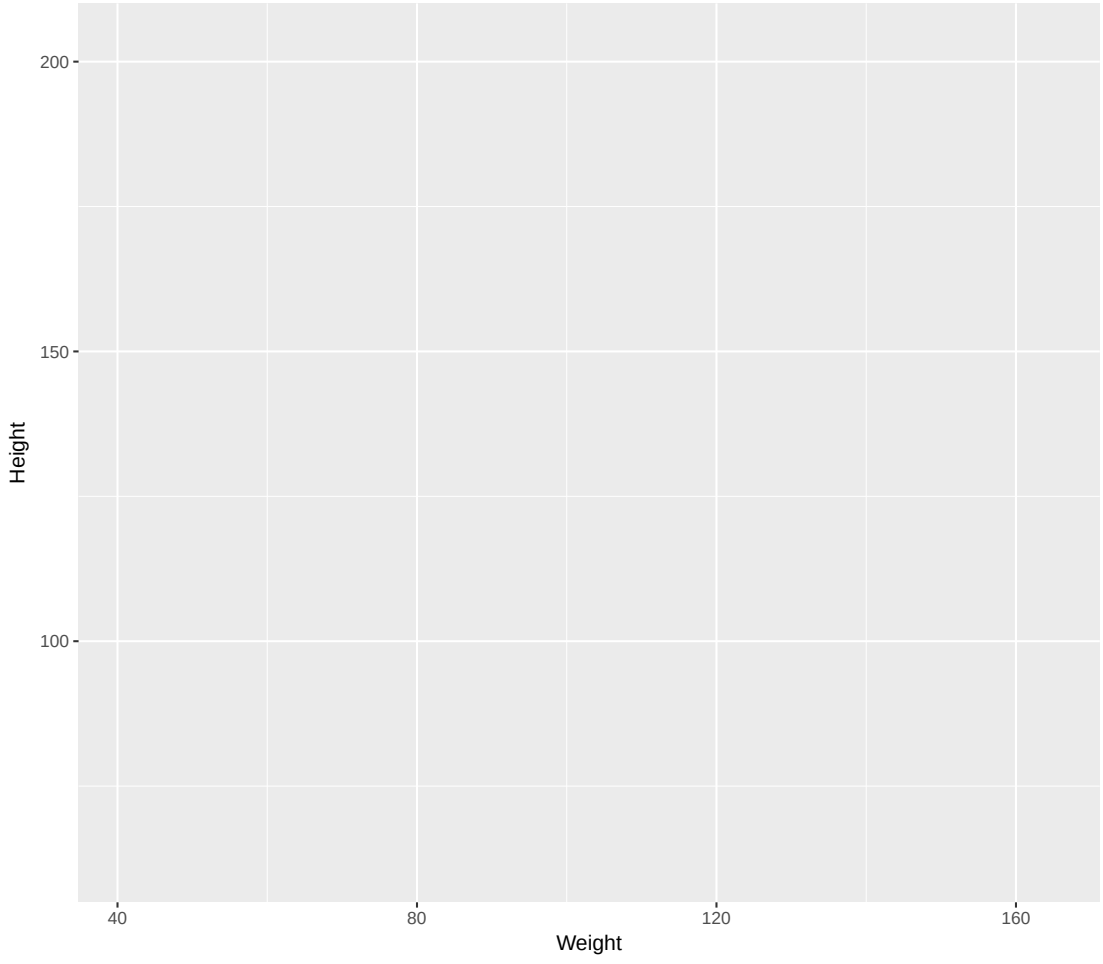
Components of plot language

`ggplot`	creates new plot (and loads data)
`aes`	defines aesthetic mappings of variables
`geom_...`	defines a layer of geometric objects (lines, bars, etc.)
`scale_...`	controls how values are translated to visual properties of displayed objects
`coord_...`	determines how values of `x` and `y` are translated to positions in the plot
`facet_...`	generates multiple small plots
`annotate_...`	creates a separate layer of annotations
`+`	combines components of a plot
`ggsave`	saves a plot to a file

Example of layered plot

1. Data and main aesthetics layer

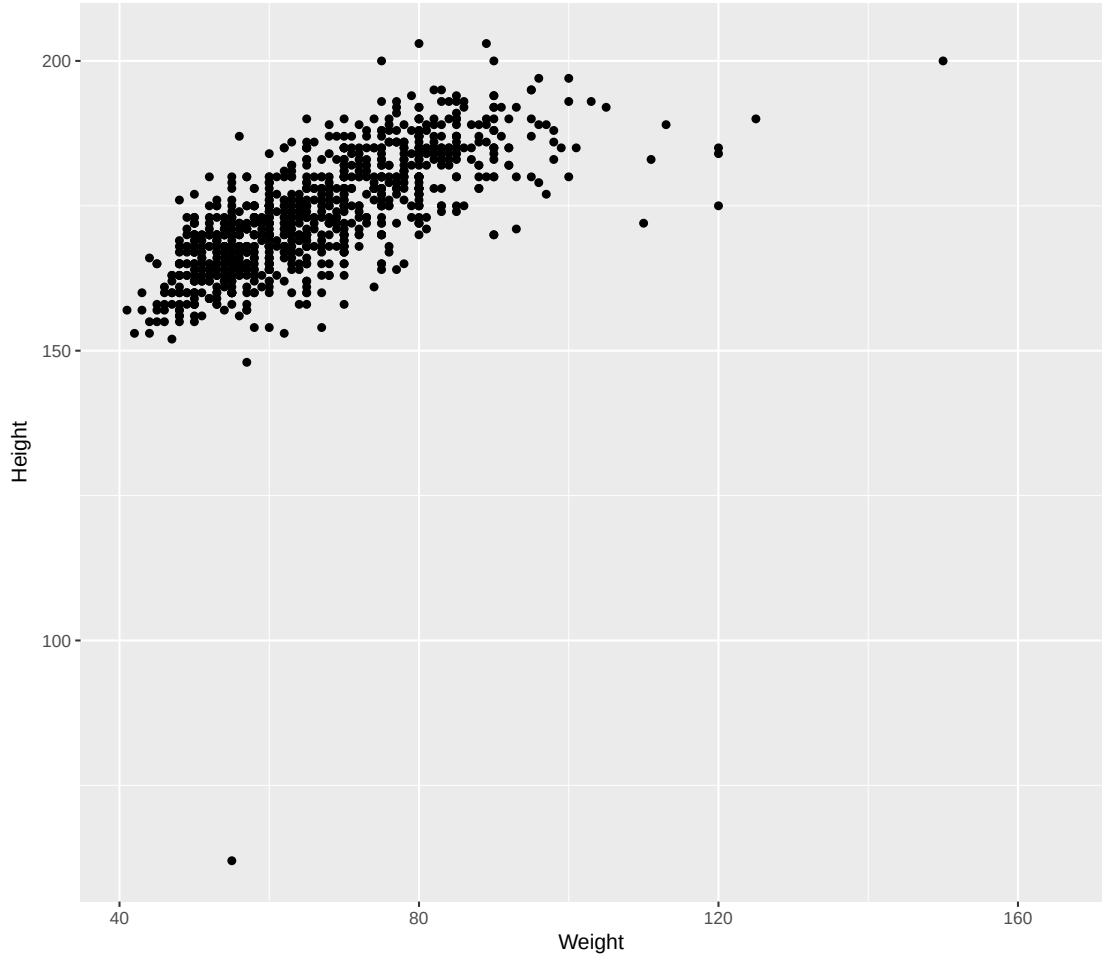
```
ggplot(df,aes(x=Weight,y=Height))
```



Example of layered plot

2. Geometry layer

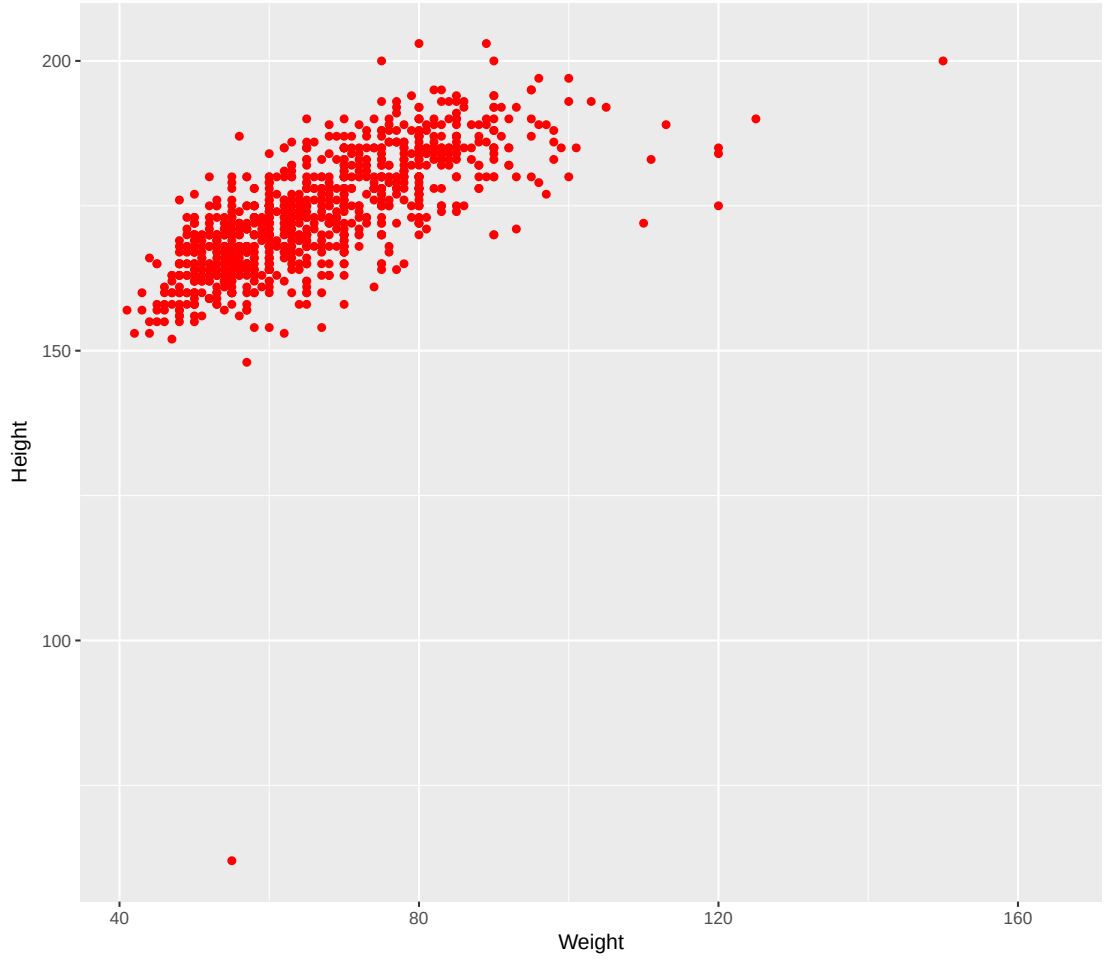
```
ggplot(df,aes(x=Weight,y=Height)) +
geom_point()
```



Example of layered plot

2. Parameterized geometry layer

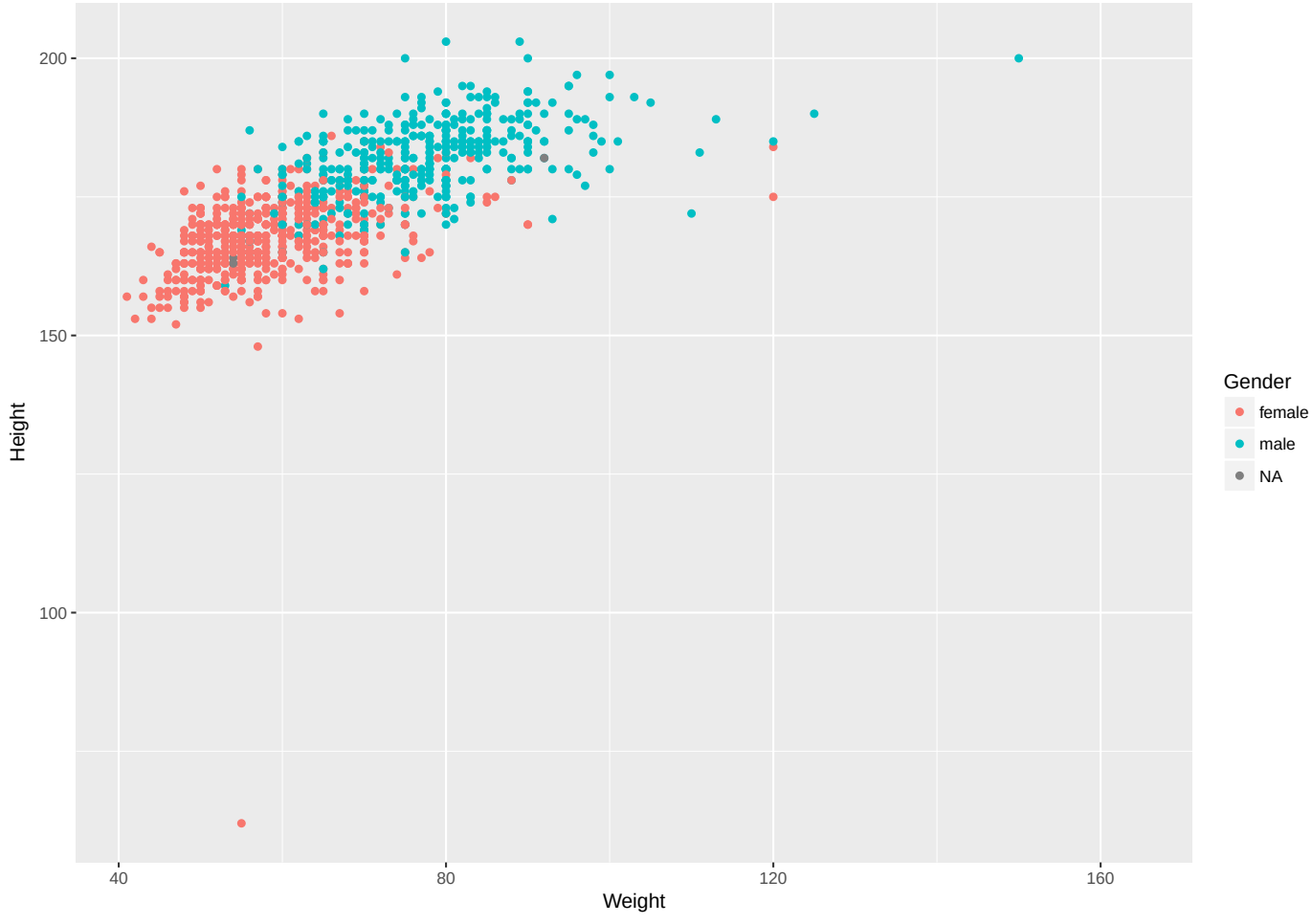
```
ggplot(df,aes(x=Weight,y=Height)) +
geom_point(color="red")
```



Example of layered plot

2. Variable geometry layer

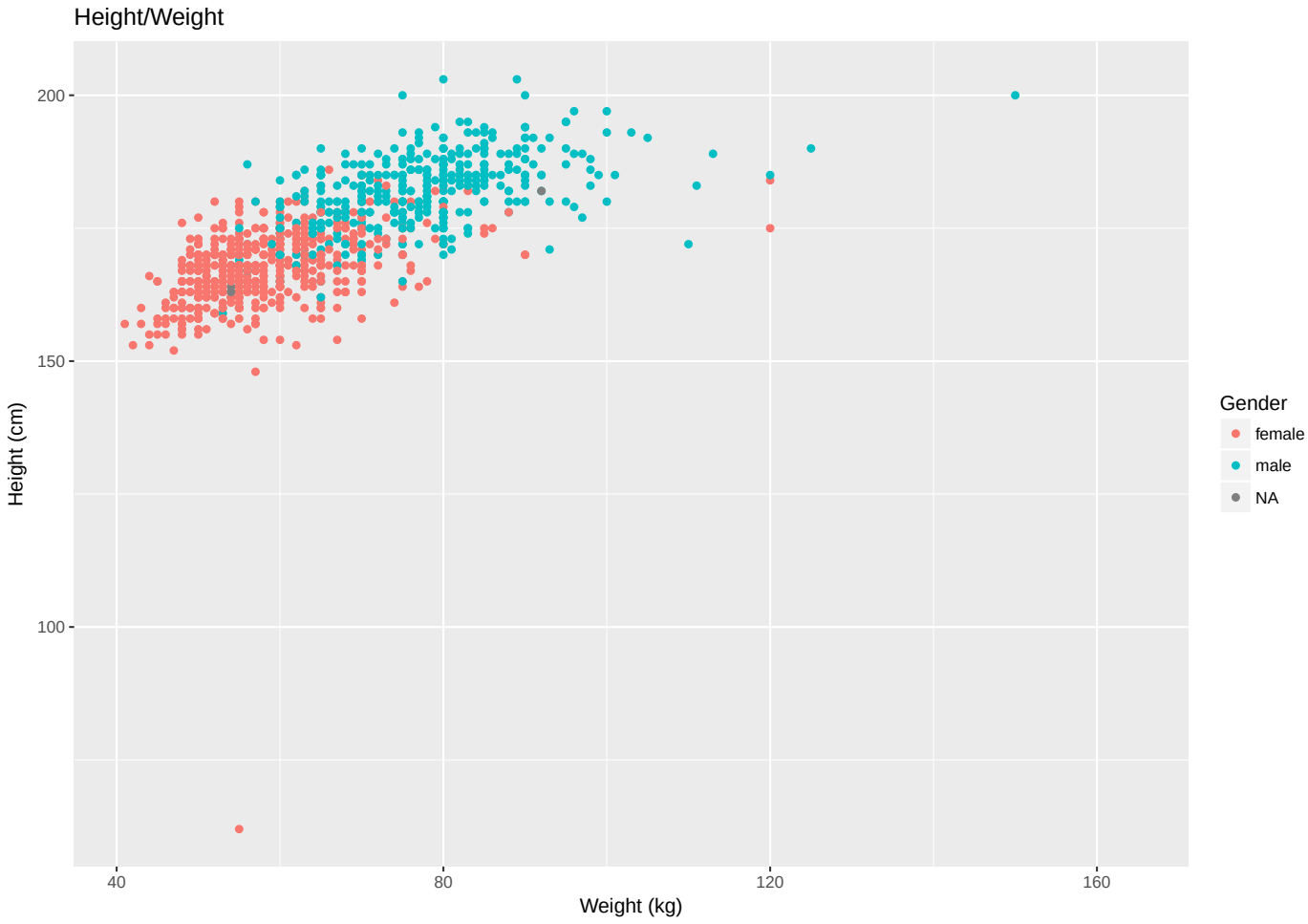
```
ggplot(df,aes(x=Weight,y=Height)) +
geom_point(aes(color=Gender))
```



Example of layered plot

3. Label/title layer

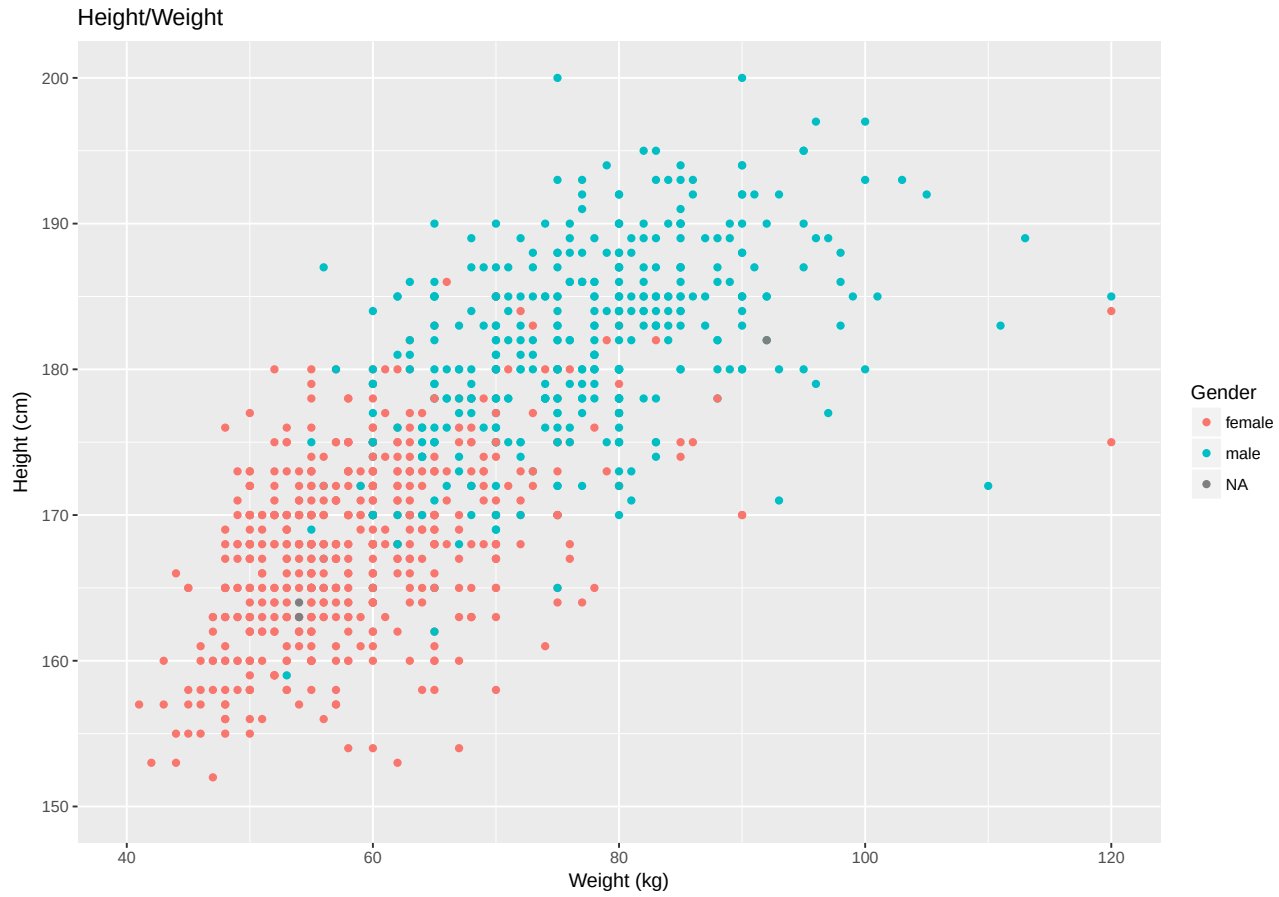
```
ggplot(df,aes(x=Weight,y=Height)) +
geom_point(aes(color=Gender)) +
labs(title="Height/Weight",x="Weight (kg)",y="Height (cm)")
```



Example of layered plot

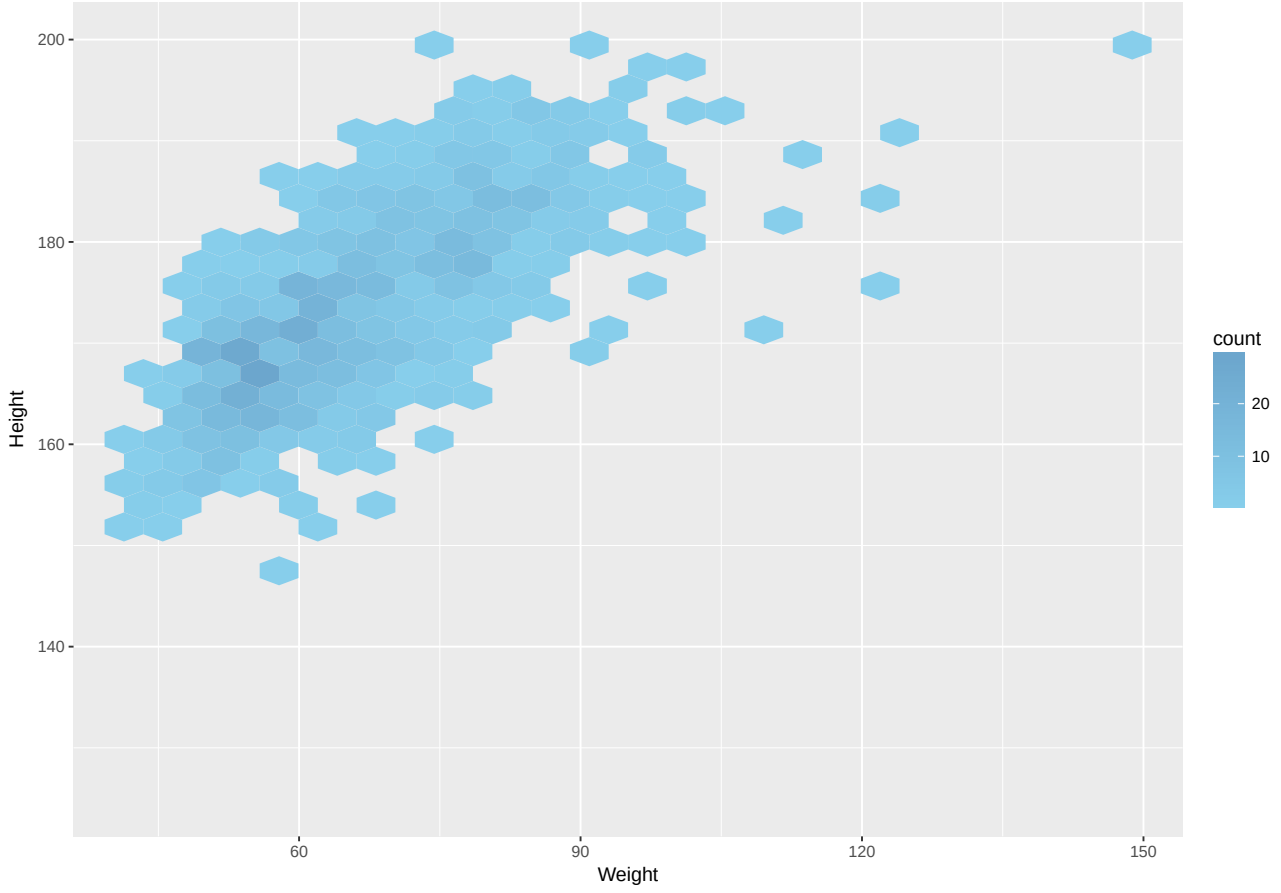
4. Data filter/manipulation layer

```
ggplot(df,aes(x=Weight,y=Height)) +
geom_point(aes(color=Gender)) +
labs(title="Height/Weight",x="Weight (kg)",y="Height (cm)") +
coord_cartesian(xlim=c(40,120), ylim=c(150,200))
```



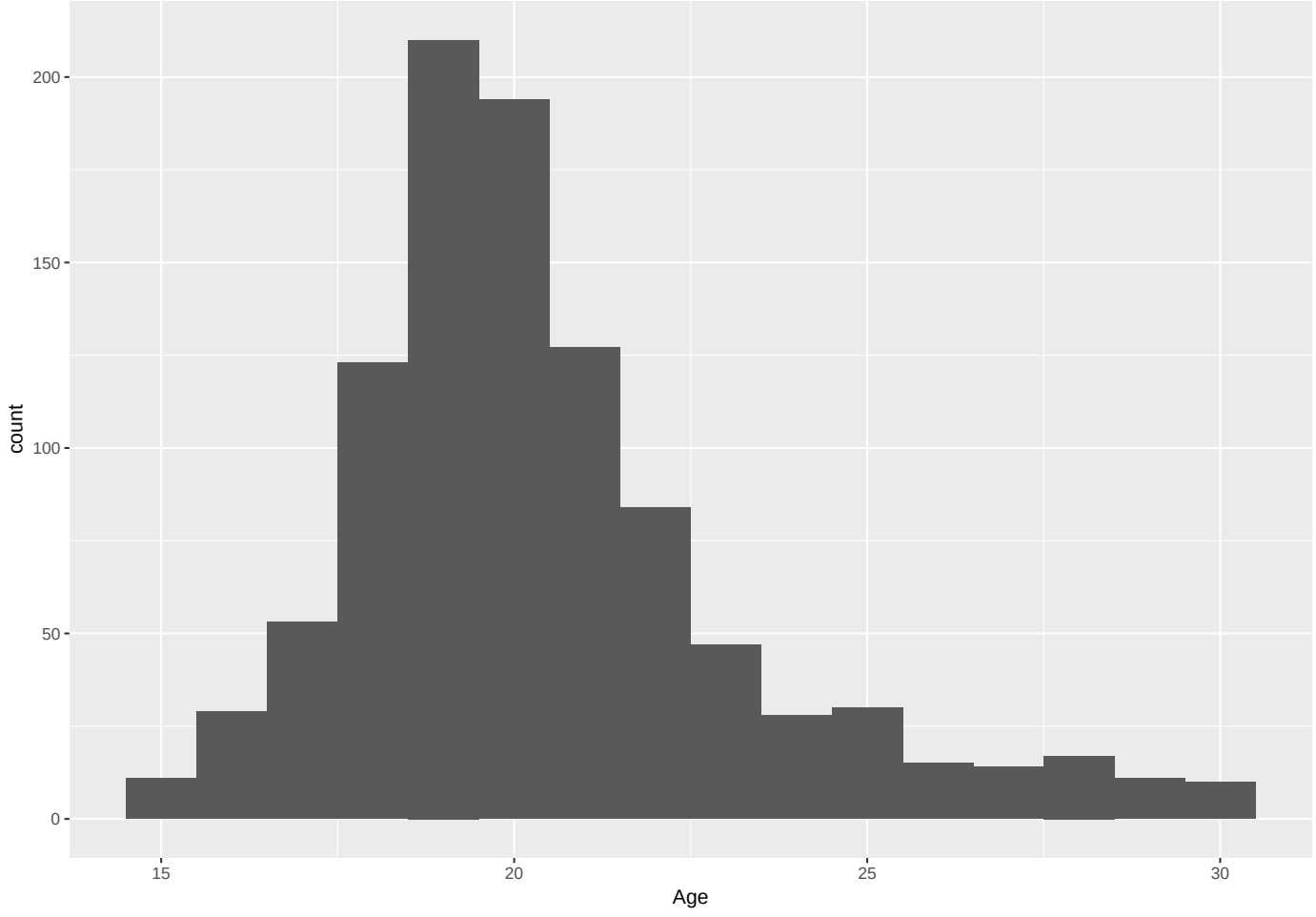
Honeycomb density plots

```
ggplot(df,aes(x=Weight,y=Height)) + geom_hex() +
lims(y=c(125,200)) +
scale_fill_gradientn(colors=c("skyblue","skyblue3"))
```



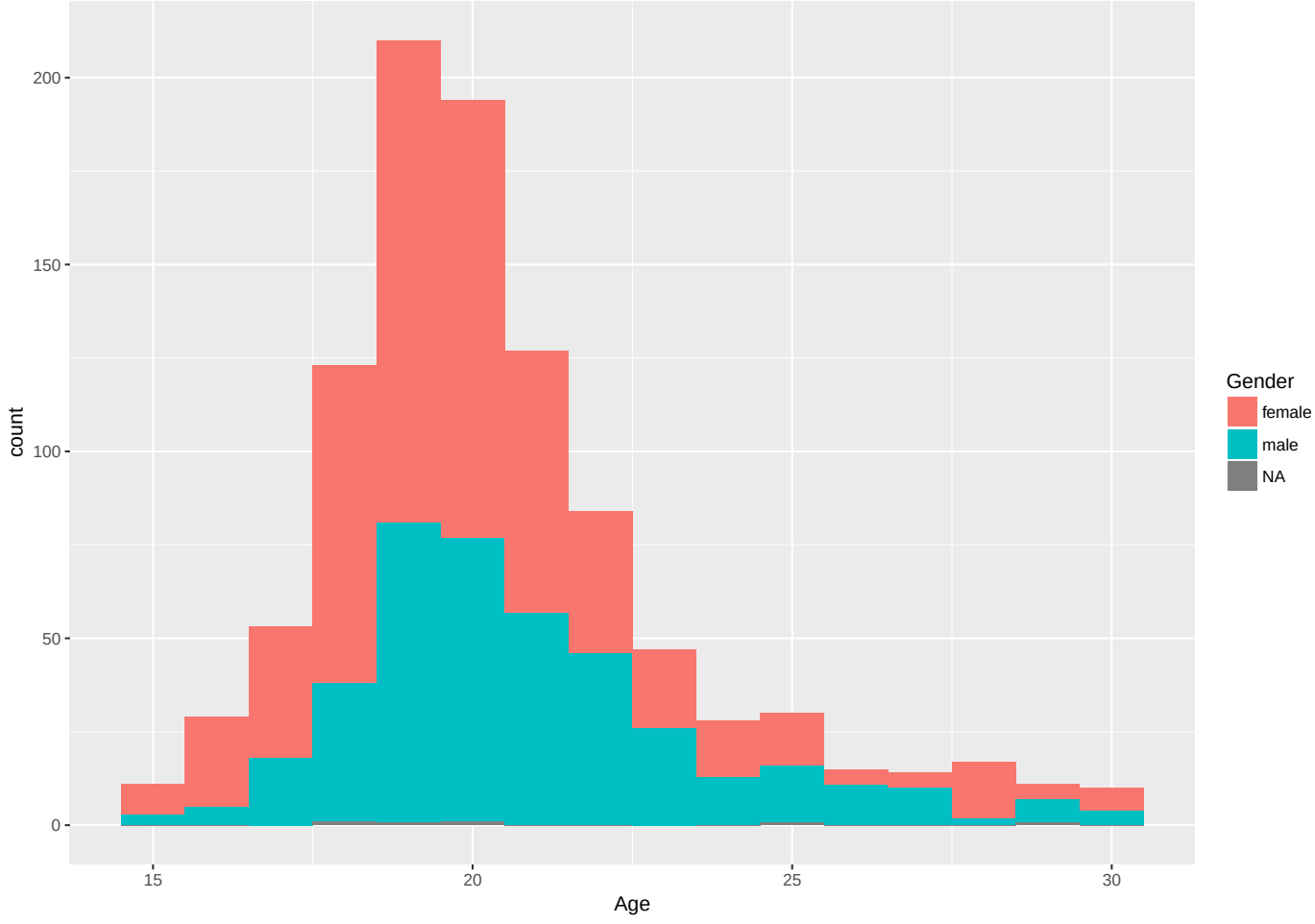
Histograms

```
ggplot(df,aes(x=Age)) +
geom_histogram(binwidth=1)
```



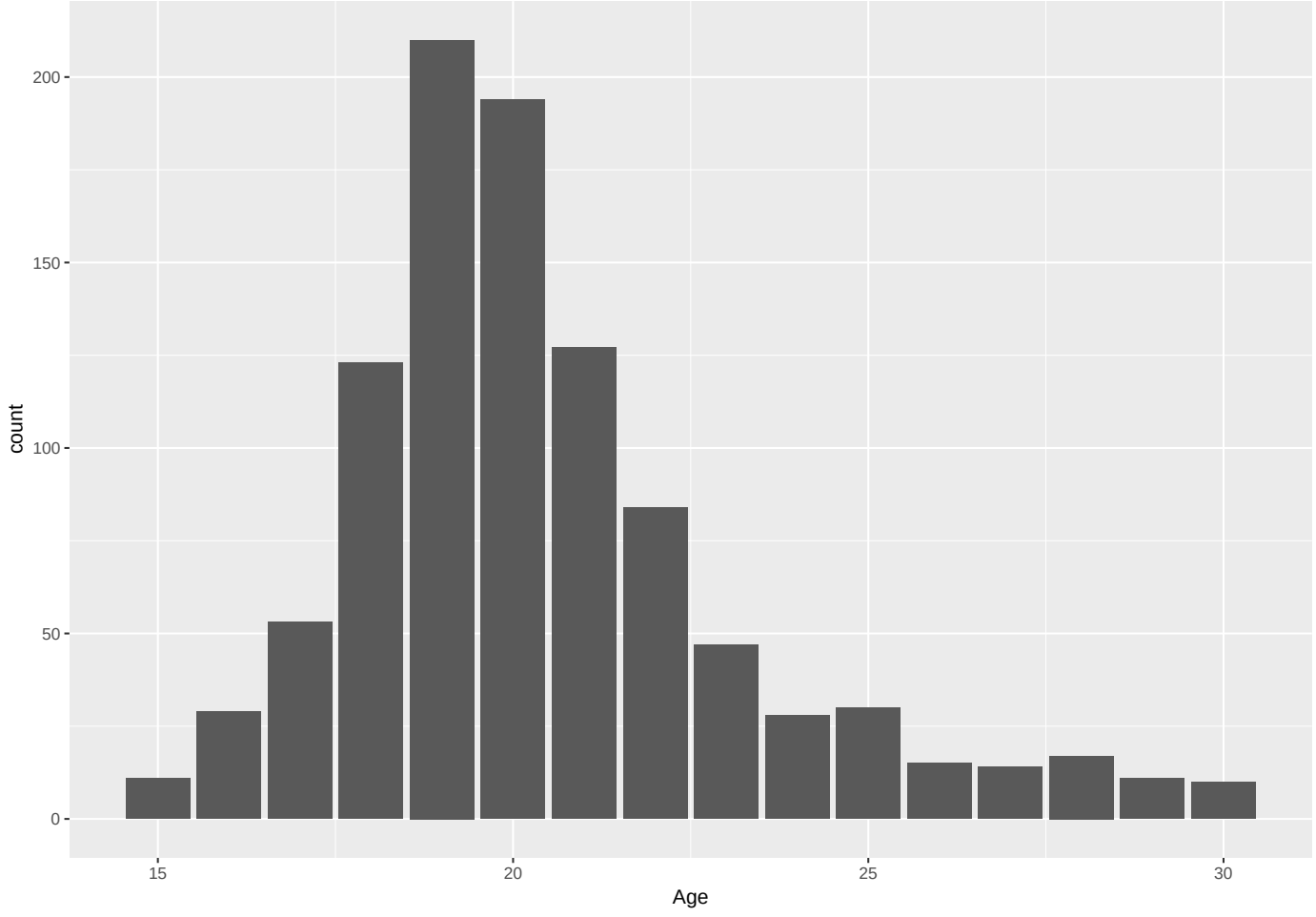
Histograms

```
ggplot(df,aes(x=Age)) +
geom_histogram(aes(fill=Gender),binwidth=1)
```



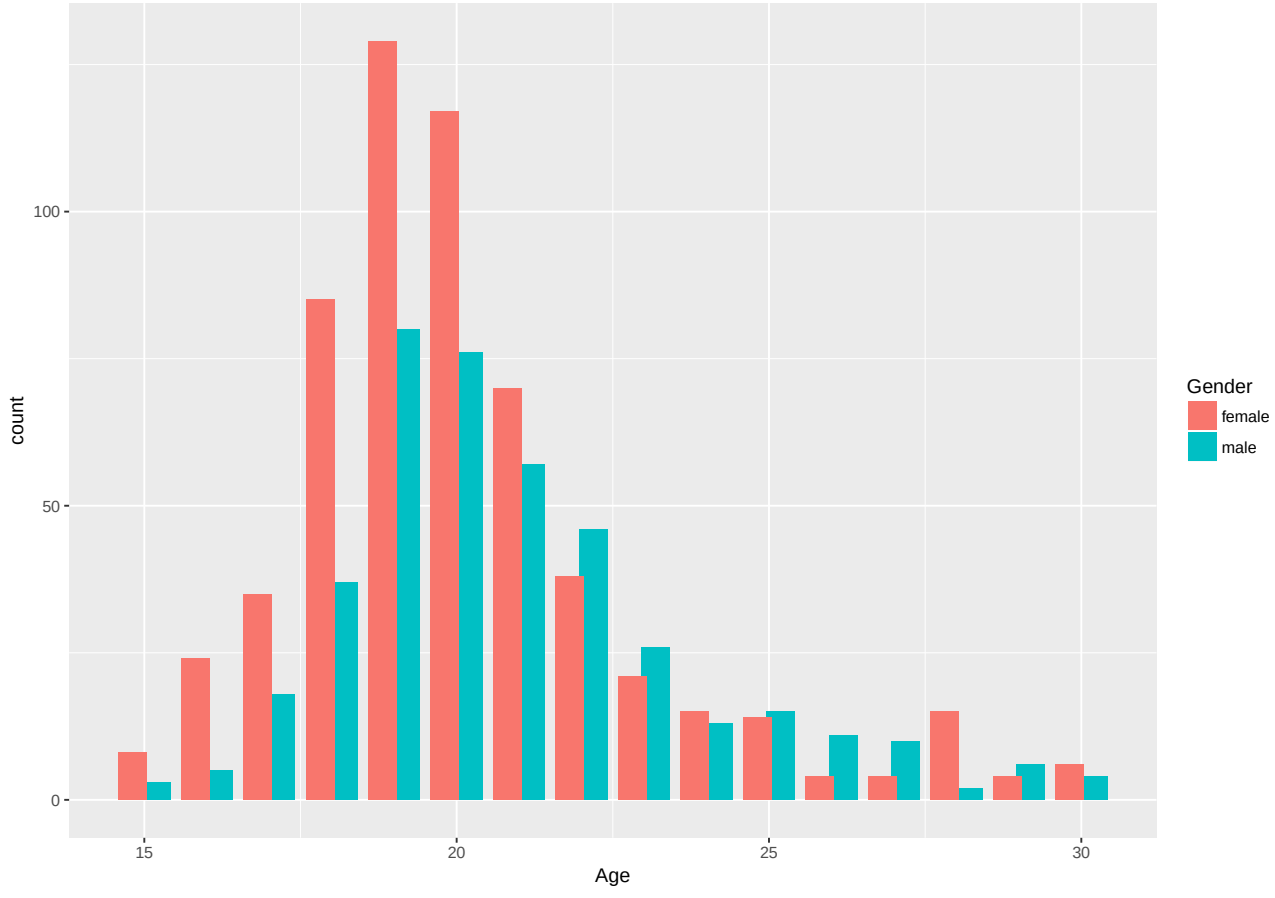
Histograms

```
ggplot(df,aes(x=Age)) +
geom_bar()
```



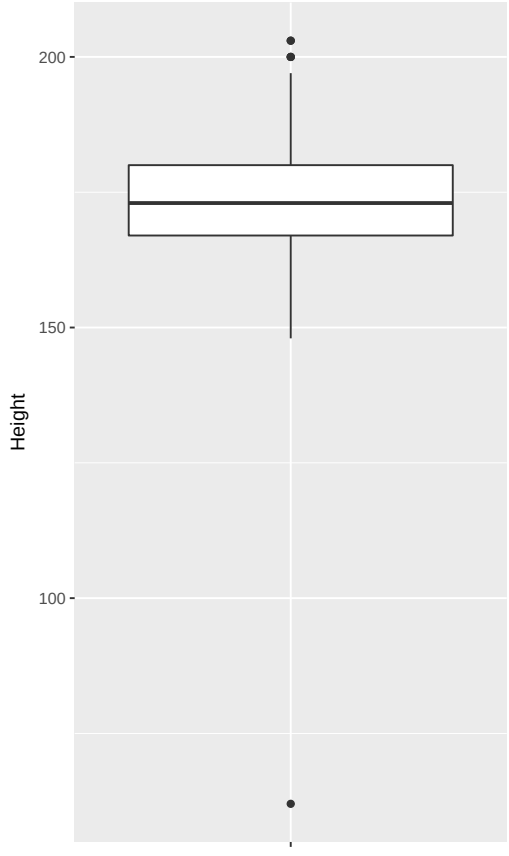
Histograms

```
ggplot(df[!is.na(df$Gender),],aes(x=Age)) +
geom_bar(aes(fill=Gender),position=position_dodge(width=0.75))
```



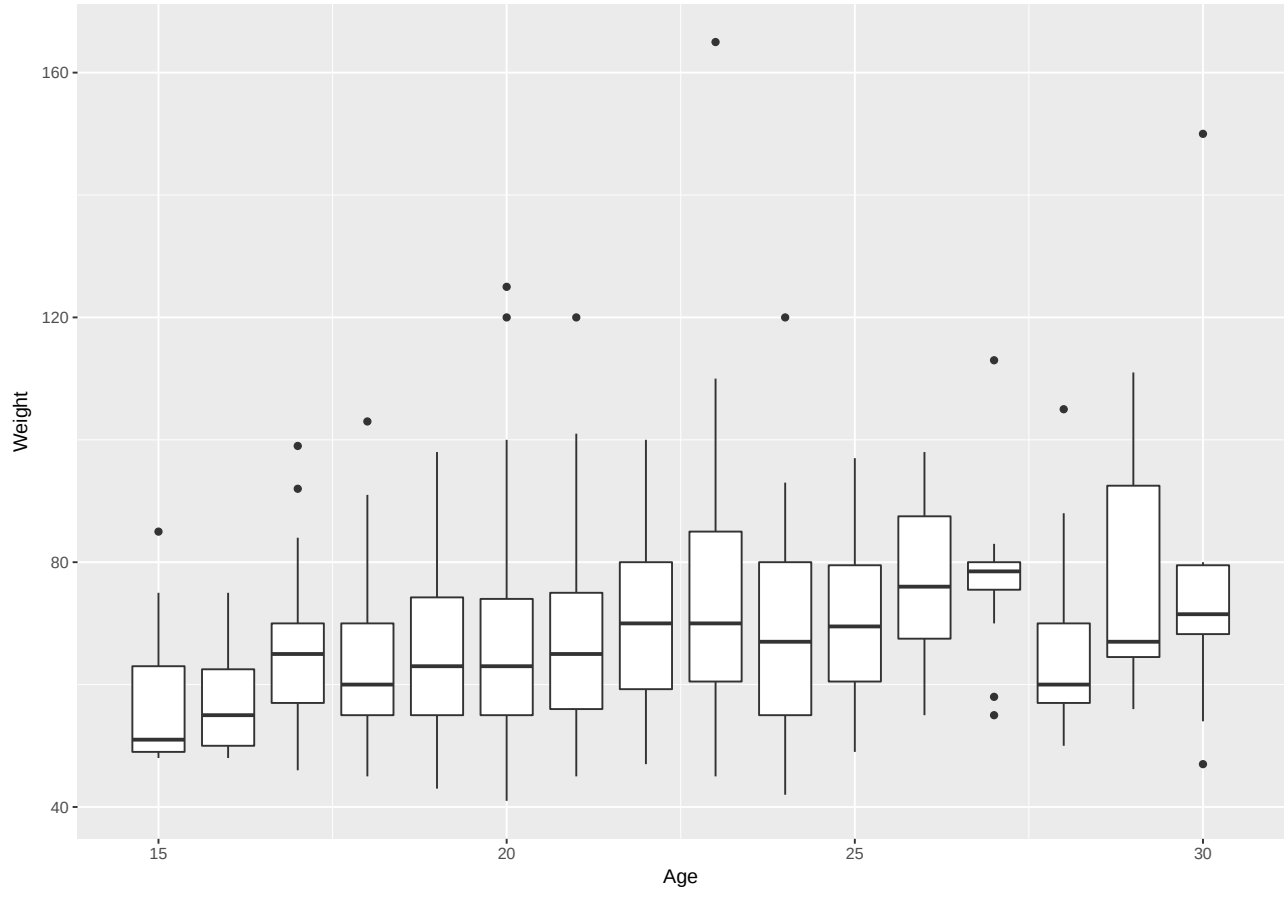
Whisker plots

```
ggplot(df,aes(x=' ',y=Height)) + geom_boxplot()
```



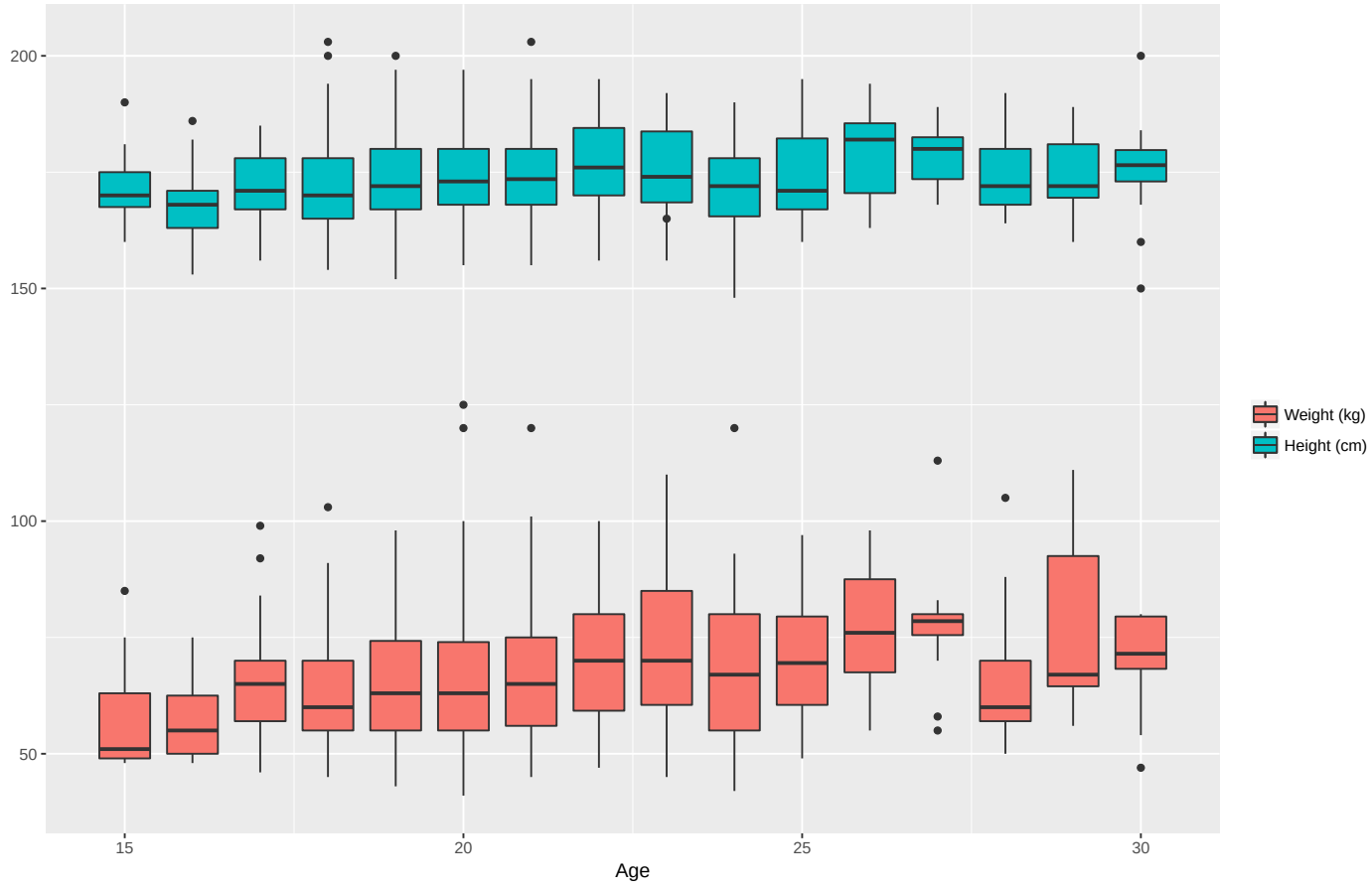
Whisker plots

```
ggplot(df,aes(x=Age)) +
geom_boxplot(aes(y=Weight,group=Age))
```



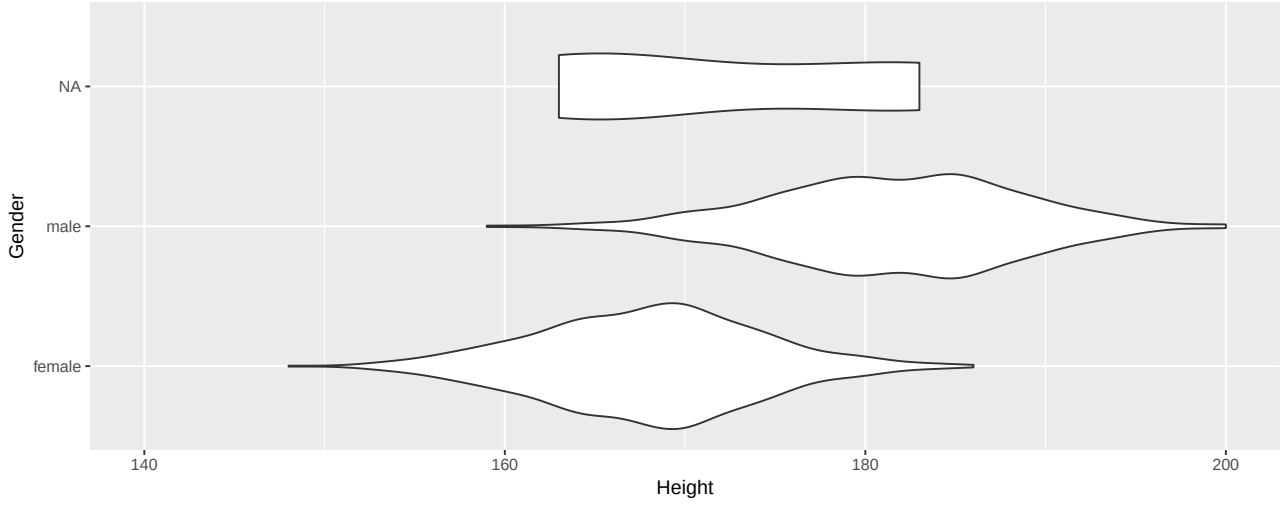
Whisker plots

```
ggplot(df,aes(x=Age)) + labs(y='') +
geom_boxplot(aes(fill='green',y=Weight,group=Age)) +
geom_boxplot(aes(fill='magenta',y=Height,group=Age)) +
scale_fill_discrete(name='',labels=c('Weight (kg)','Height (cm)'))
```



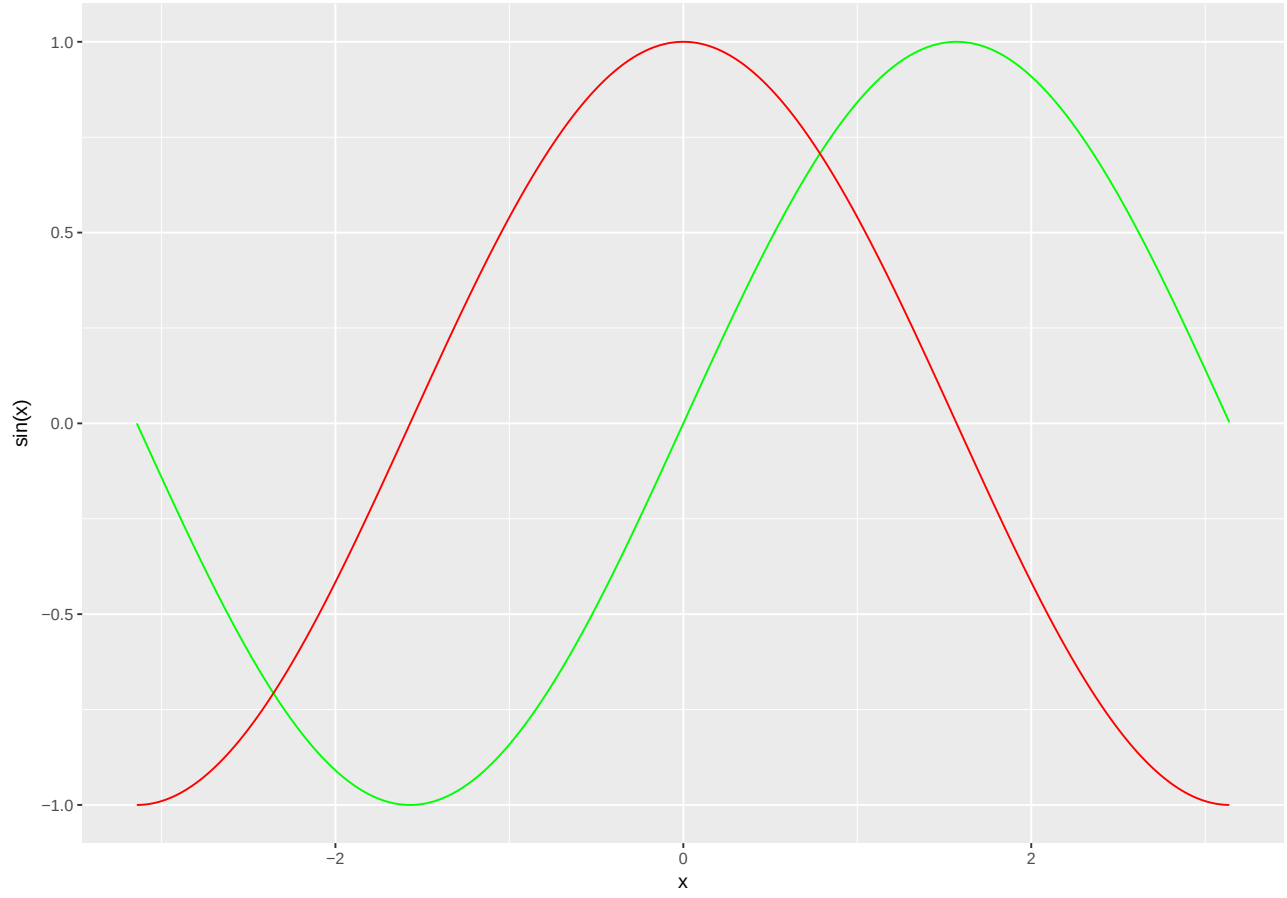
Violin (density) plots

```
ggplot(df,aes(x=Gender)) +
geom_violin(aes(y=Height)) +
lims(y=c(140,200)) + coord_flip()
```



Line plots

```
ggplot(data.frame(x=seq(-pi,pi,0.01)),aes(x)) +
geom_line(color='green',aes(y=sin(x)))+
geom_line(color='red',aes(y=cos(x)))
```

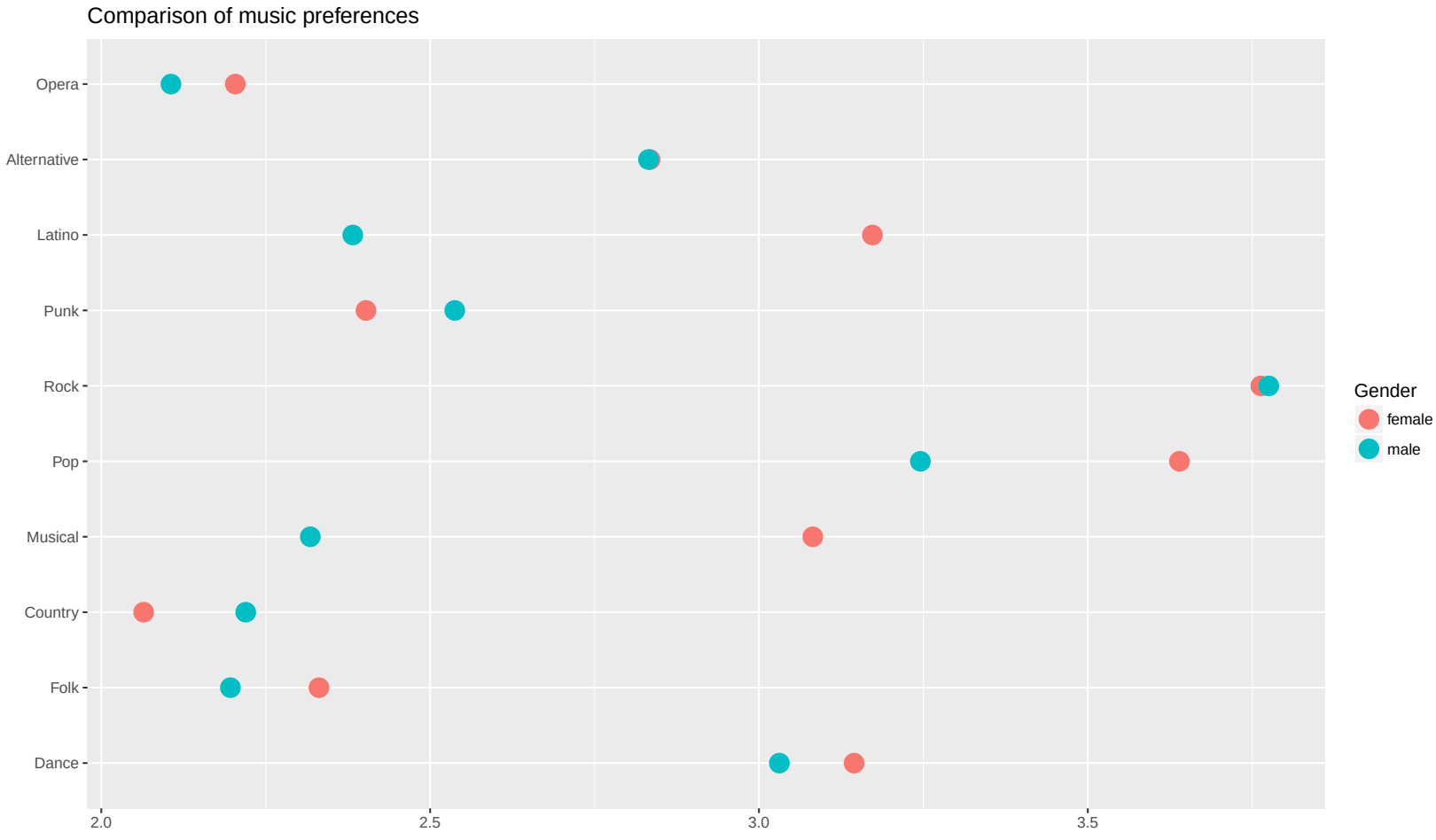


Interpolations plots

```
ggplot(df[!is.na(df$Gender),],aes(x=Weight,y=Height)) +
geom_point(aes(color=Gender)) +
geom_smooth(aes(color=Gender)) + lims(y=c(150,210))
```

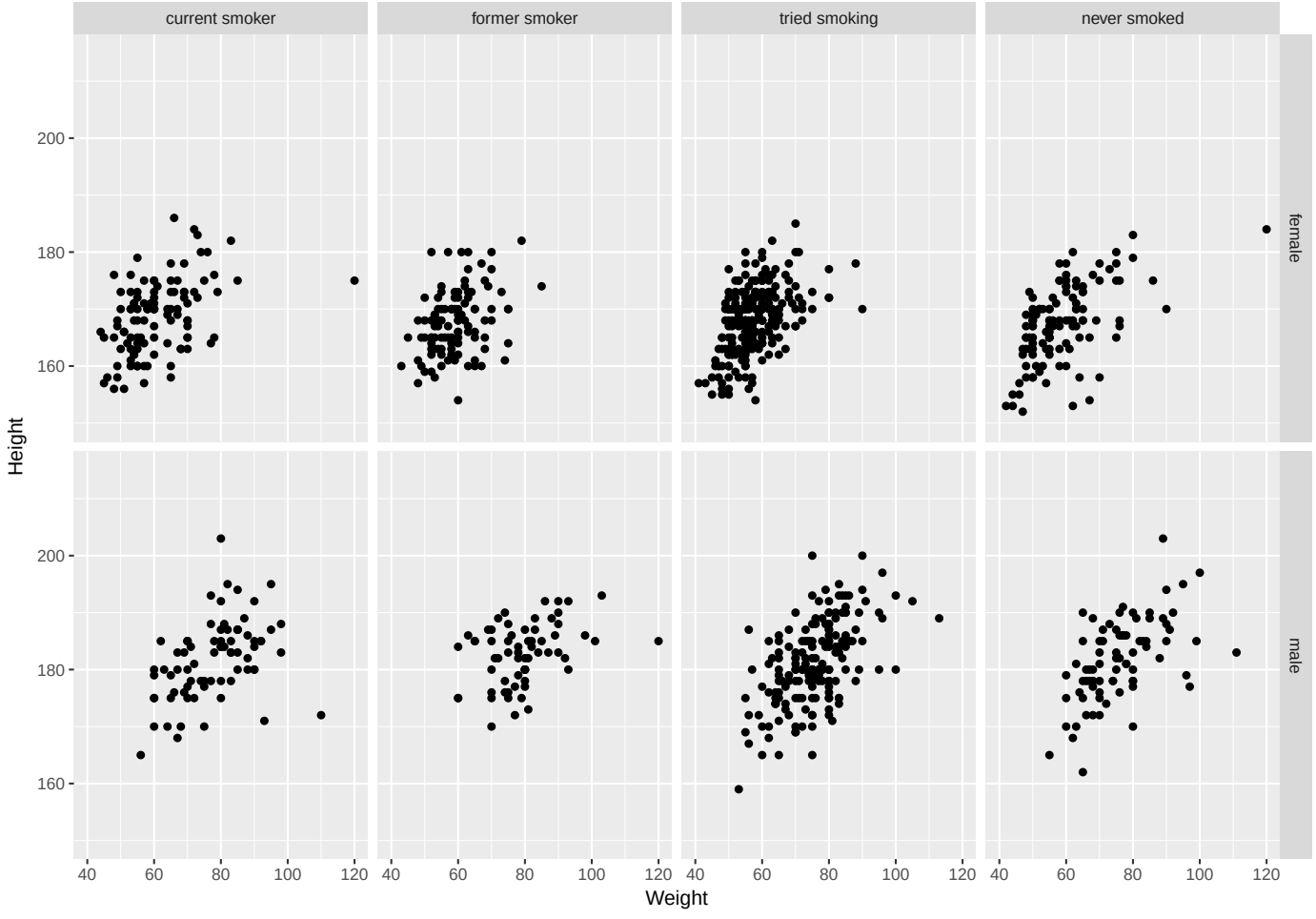
Scatter plots

```
ggplot(melt(aggregate(cbind(Dance,Folk,Country,...)~Gender,df,mean)),
    aes(x=value,y=variable)) +
geom_point(size=5,aes(color=Gender)) +
labs(title="Comparison of music preferences",x=' ',y=' ')
```



Faceting plots

```
ggplot(df[!is.na(df$Gender)&!is.na(df$Smoking),],
    aes(x=Weight,y=Height)) +
geom_point() + lims(x=c(40,120),y=c(150,215)) +
facet_grid(Gender~Smoking)
```



Saving plots to a file

- `ggsave("filename.ext")` saves the last plot to the given file name in the format indicated with the file extension
- supported file formats are "eps", "ps", "tex" (pictex), "pdf", "jpeg", "tiff", "png", "bmp", and "svg".
- the width, height, and resolution can be additionally specified with parameters

Example

```
p ← ggplot(df,aes(x=Weight,y=Height)) +
geom_point(aes(color=Gender))

ggsave('plot.png',plot=p,width=10,height=7)
```