

NZSSN Courses: Introduction to R

Session 6 – Advanced Graphics

Statistical Consulting Centre

consulting@stat.auckland.ac.nz
The Department of Statistics
The University of Auckland

20 July, 2017



SCIENCE
DEPARTMENT OF STATISTICS

ggplot2 package

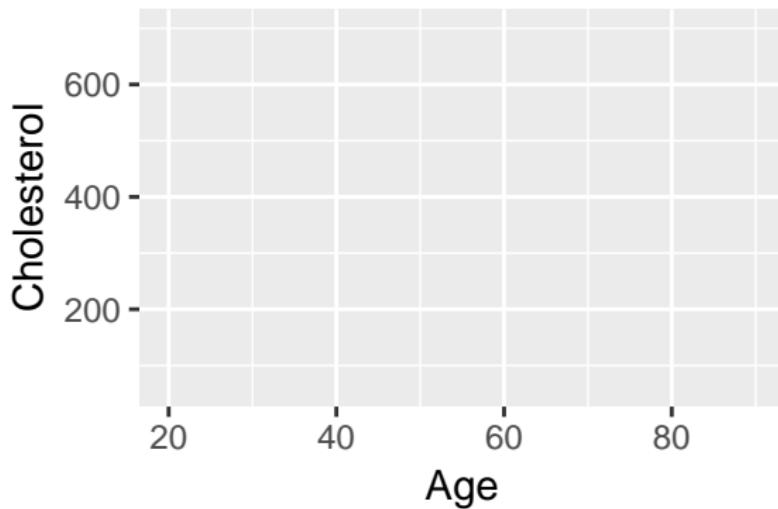
- Documentation: <http://docs.ggplot2.org/current/>
- recommended reading “The Layered Grammar of Graphics”:
<http://vita.had.co.nz/papers/layered-grammar.pdf>
- Load ggplot2 package

```
library(ggplot2)
```

Create a new ggplot

- Initialising a ggplot object.

```
ggplot(data = combined.long.df,  
       mapping = aes(x = Age, y = Cholesterol))
```



Create a new ggplot

- Initialising a ggplot object.

```
ggplot(data = combined.long.df,  
       mapping = aes(x = Age, y = Cholesterol))
```

There are three common ways to invoke ggplot:

- `ggplot(combined.long.df, aes(x, y, <other aesthetics>))`
- `ggplot(combined.long.df)`
- `ggplot()`

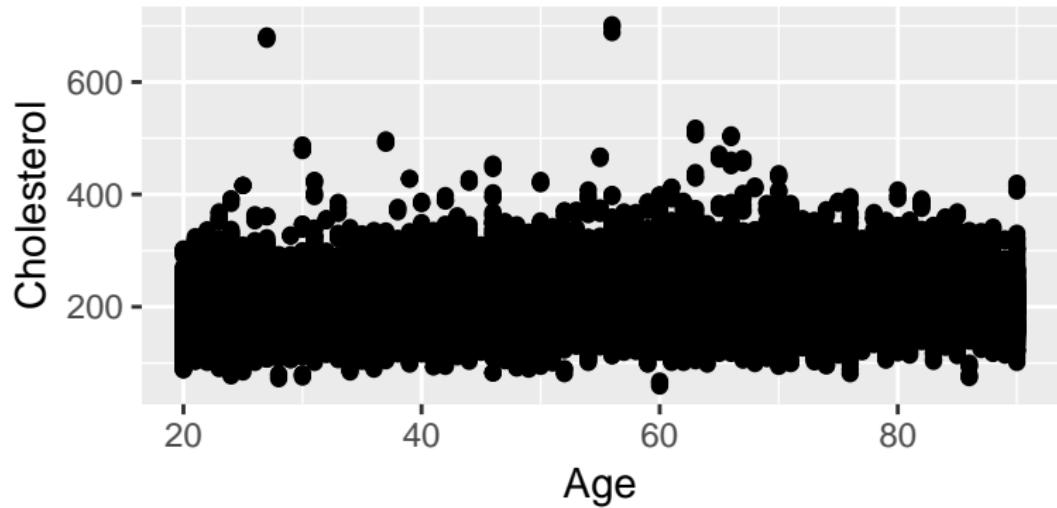
Create a new ggplot

- assign this ggplot object to a variable

```
g <- ggplot(data = combined.long.df,  
             mapping = aes(x = Age, y = Cholesterol))
```

Create a Scatterplot

```
g + geom_point()
```



- `geom`, short for geometric object, describes the type of plot you will produce.

Create a Scatterplot

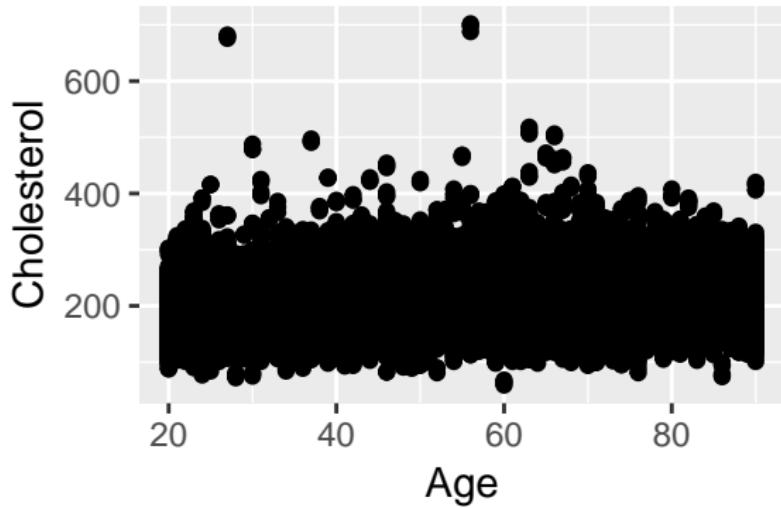
- Note that here are three common ways to invoke ggplot:

```
ggplot(data = combined.long.df,  
       mapping = aes(x = Age, y = Cholesterol)) + geom_point()  
ggplot(data = combined.long.df) +  
  geom_point(mapping = aes(x = Age, y = Cholesterol))  
ggplot() +  
  geom_point(data = combined.long.df,  
             mapping = aes(x = Age, y = Cholesterol))
```

- always check the documentation, `?geom_point`, for which aesthetics can be used.

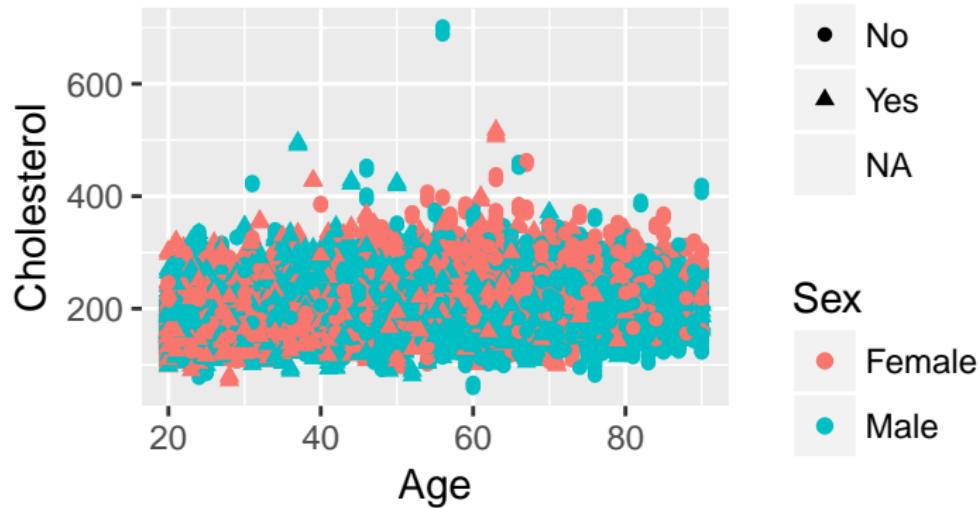
First method is recommended

```
g <- ggplot(data = combined.long.df,  
             mapping = aes(x = Age, y = Cholesterol))  
g + geom_point()
```



Control colour and shape

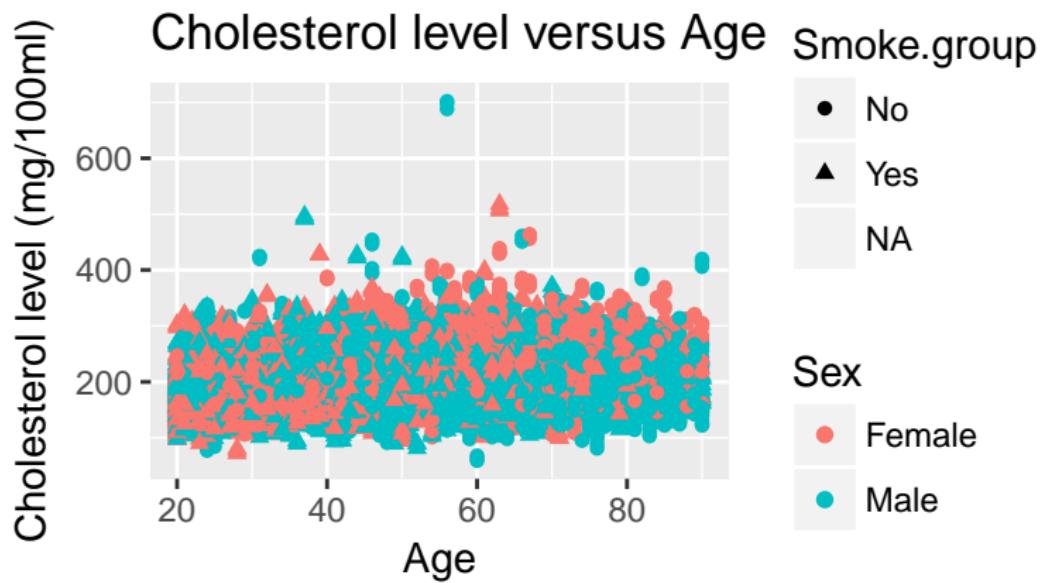
```
g + geom_point(aes(colour = Sex, shape = Smoke.group))
```



- always check the documentation, `?geom_point`, for which aesthetics can be used.
- note the missing values in the legend labelling

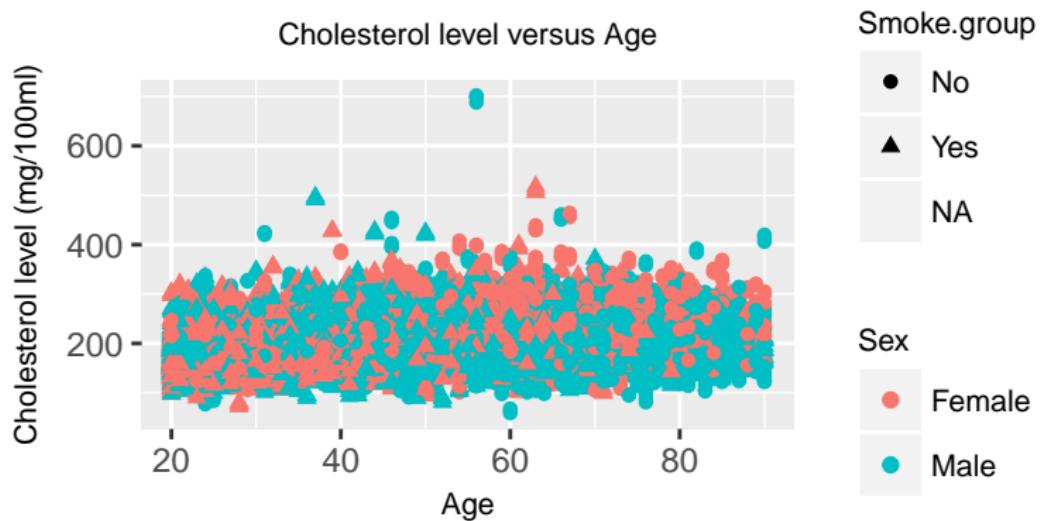
Modify axis, legend, and plot labels

```
(g <- g + geom_point(aes(colour = Sex, shape = Smoke.group)) +  
  labs(title = "Cholesterol level versus Age",  
       x = "Age", y = "Cholesterol level (mg/100ml)"))
```



Theme controls non-data components of the plot

```
g + theme(plot.title = element_text(size=8, hjust = 0.5),  
          axis.title = element_text(size=8),  
          legend.title = element_text(size=8),  
          legend.text = element_text(size=8) )
```



Create a Scatterplot

```
g <- ggplot(  
  data = na.omit(combined.long.df[,  
    c("Age", "Cholesterol", "Sex", "Smoke.group")]),  
  mapping = aes(x = Age, y = Cholesterol,  
    color = Sex, shape = Smoke.group)) +  
  geom_point() +  
  labs(title = "Total score versus Age",  
    x = "Age", y = "Total score")
```

- ggplot object can be further modified.

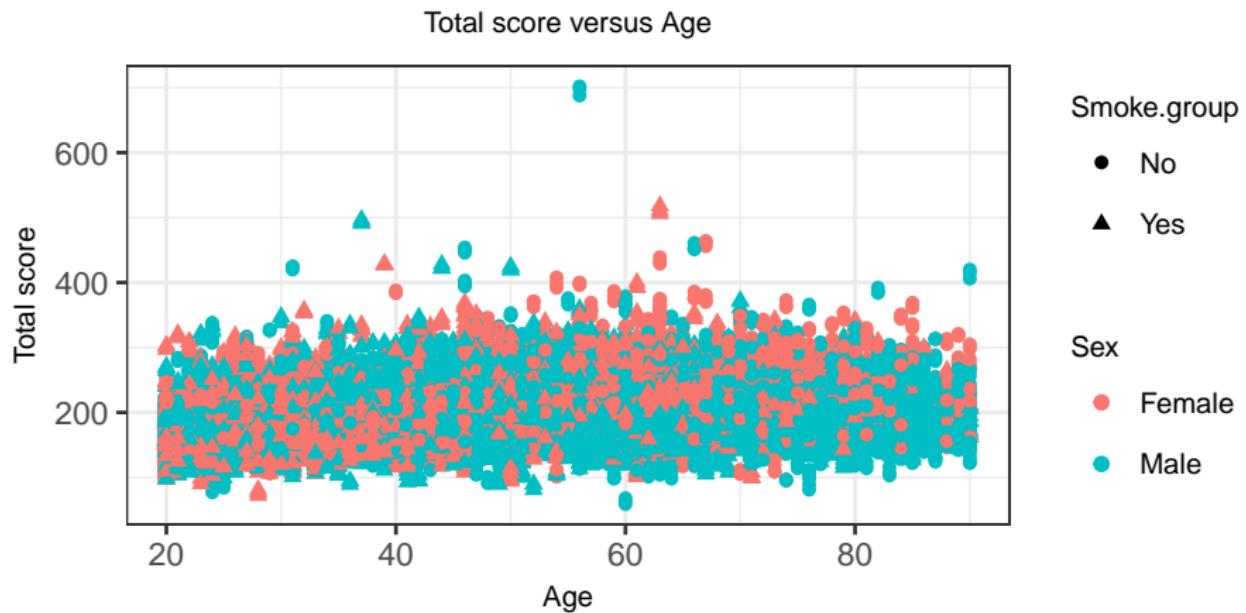
Create a Scatterplot

```
myTheme <- theme_bw() +  
  theme(plot.title = element_text(size=8, hjust = 0.5),  
        axis.title=element_text(size=8),  
        legend.title = element_text(size=8),  
        legend.text = element_text(size=8) )
```

- myTheme can be reused for different types of plot.

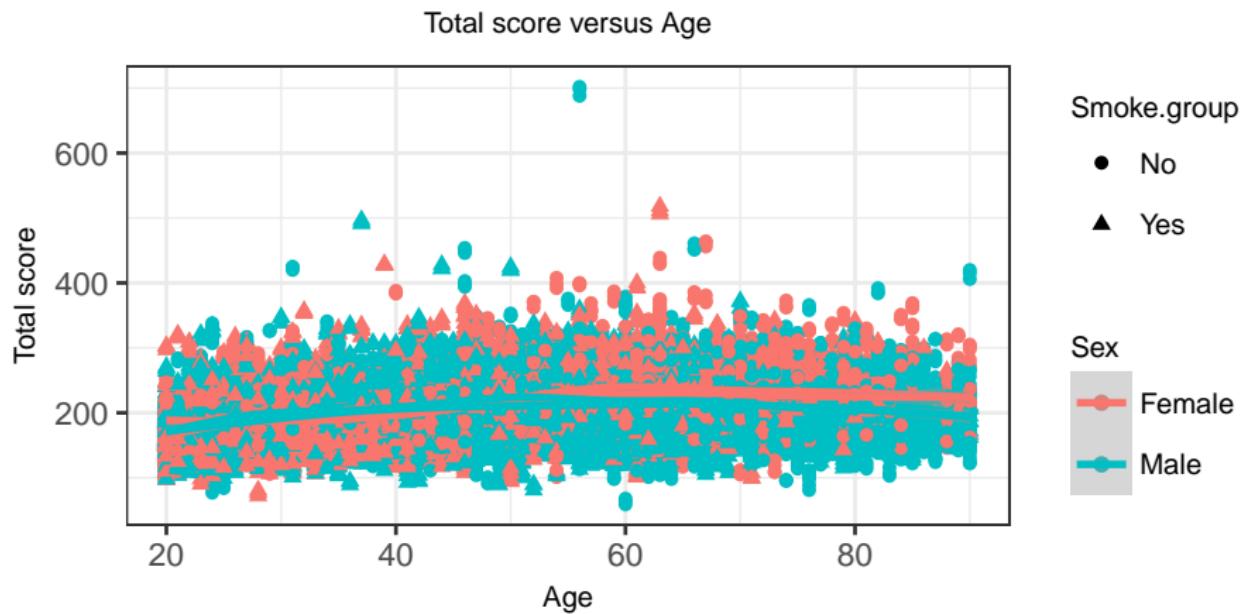
Scatterplot

g + myTheme



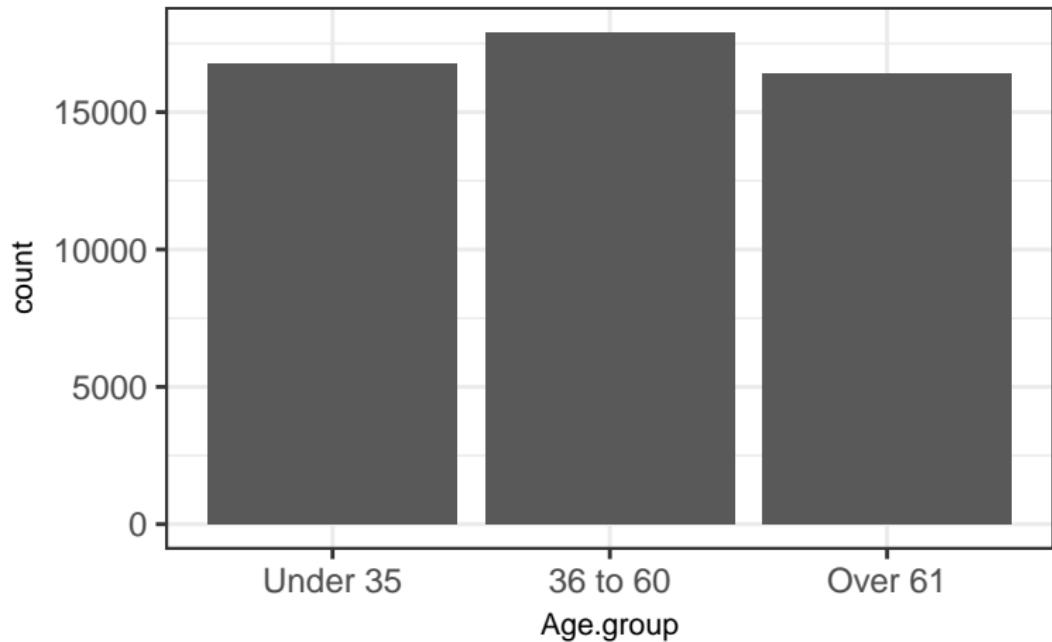
Scatterplot with a smoother

```
g + geom_smooth() + myTheme
```



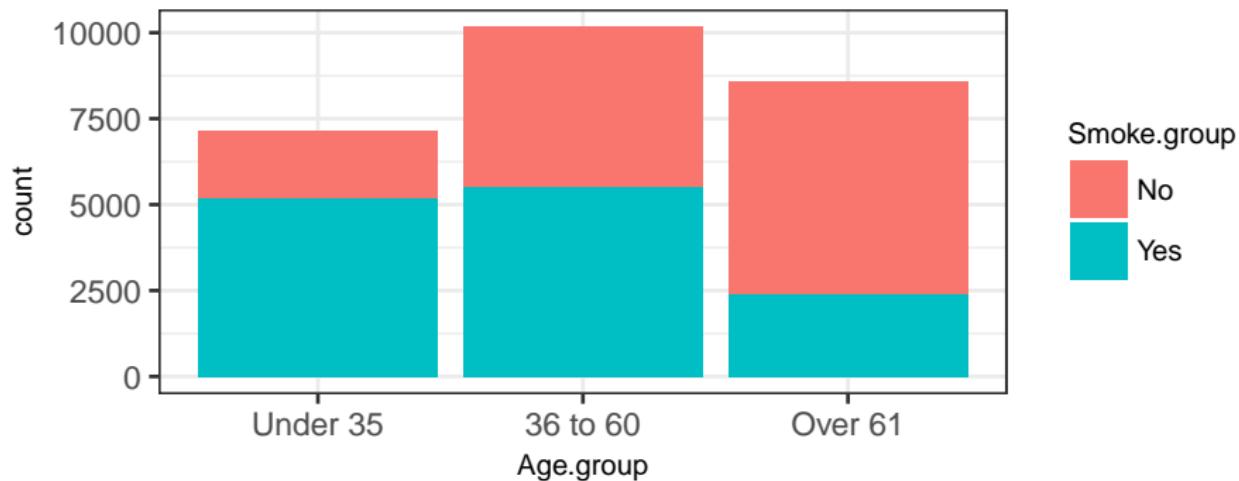
Bar chart

```
ggplot(combined.long.df, aes(x = Age.group)) + geom_bar() + my
```



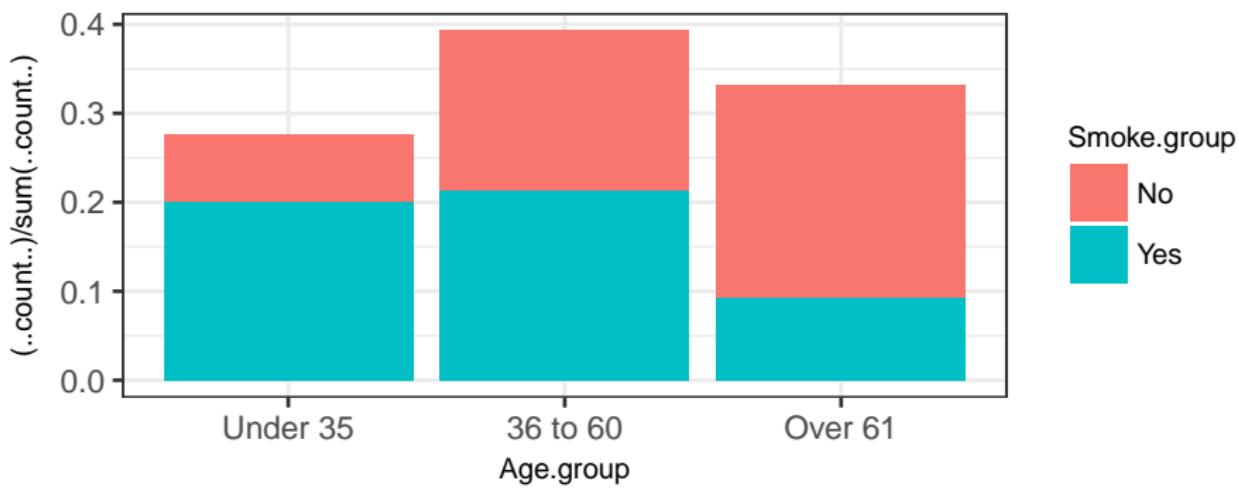
Smoking group by Age group

```
ggplot(na.omit(combined.long.df[,c("Age.group", "Smoke.group")])  
  aes(x = Age.group, fill = Smoke.group)) +  
  geom_bar() + myTheme
```



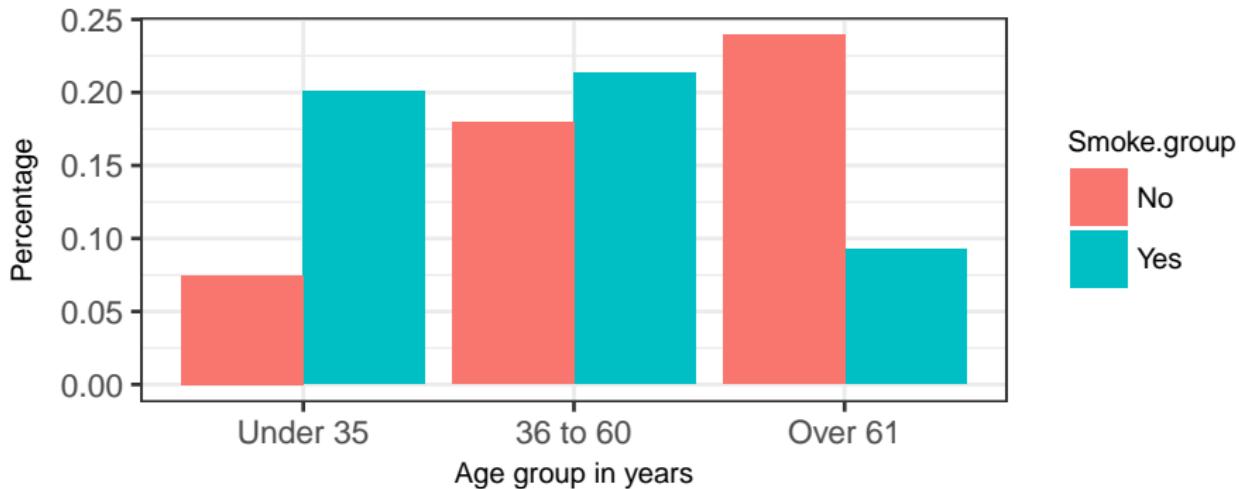
Bar chart on proportions

```
g <- ggplot(na.omit(combined.long.df[,c("Age.group", "Smoke.gr  
aes(x = Age.group, y = (.count.)/sum(.count.),  
fill = Smoke.group))  
g + geom_bar() + myTheme
```



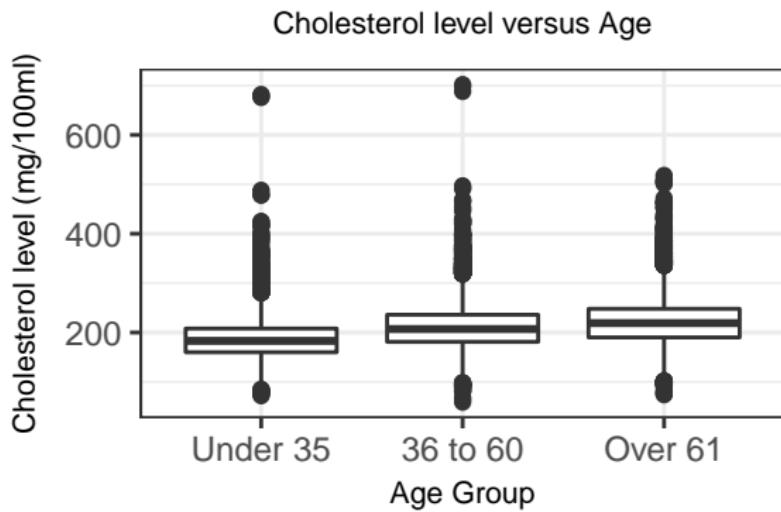
Bar chart on proportions

```
g + geom_bar(position = "dodge") +  
  labs( x = "Age group in years", y = "Percentage") +  
  myTheme
```



Boxplot using geom_boxplot()

```
ggplot(na.omit(combined.long.df[,c("Age.group", "Cholesterol")])  
      aes(x = Age.group, y = Cholesterol)) +  
  geom_boxplot() +  
  labs(title = "Cholesterol level versus Age",  
       x = "Age Group", y = "Cholesterol level (mg/100ml)") +
```

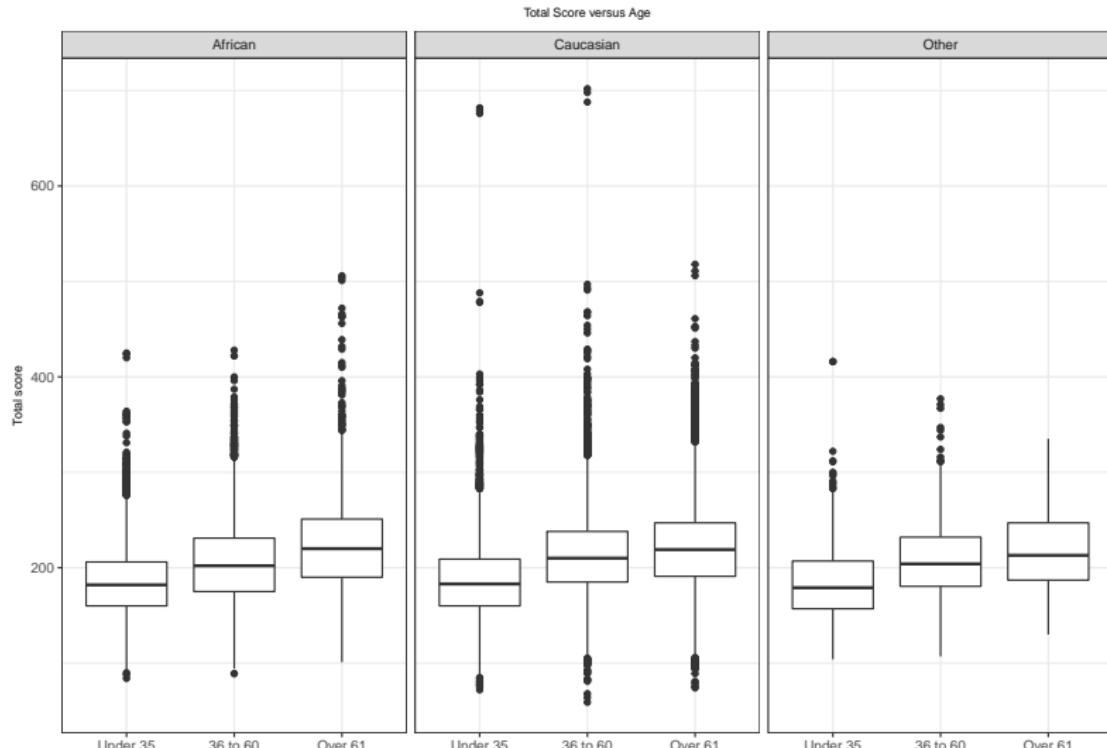


Boxplot with panels using facet_wrap()

```
g <- ggplot(na.omit(combined.long.df[,c("Age.group",
                                         "Cholesterol", "Race.group")]),
             aes(x = Age.group, y = Cholesterol)) +
  geom_boxplot() + labs(title = "Total Score versus Age",
                        x = "Age Group", y = "Total score")
```

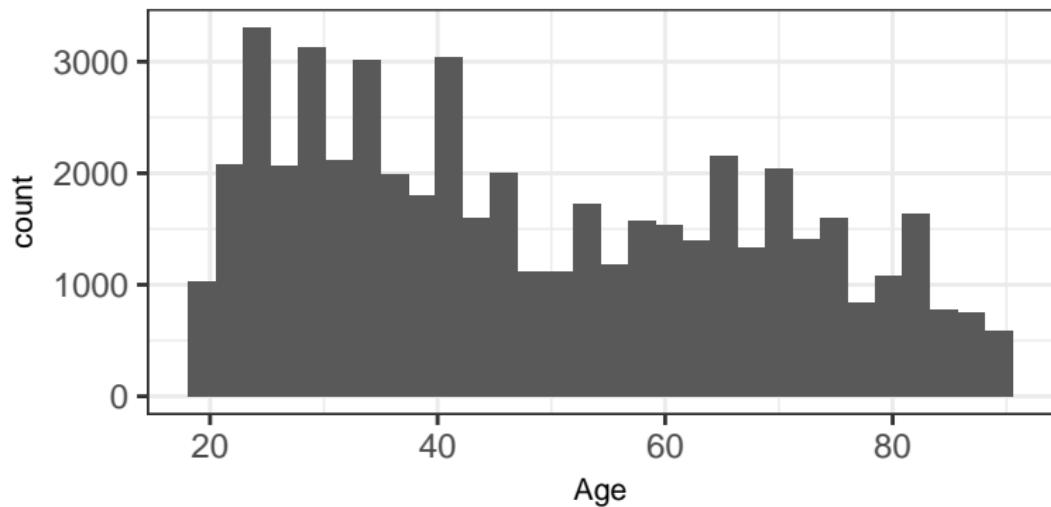
Boxplot with panels using facet_wrap()

```
g + facet_wrap(~Race.group) + myTheme
```



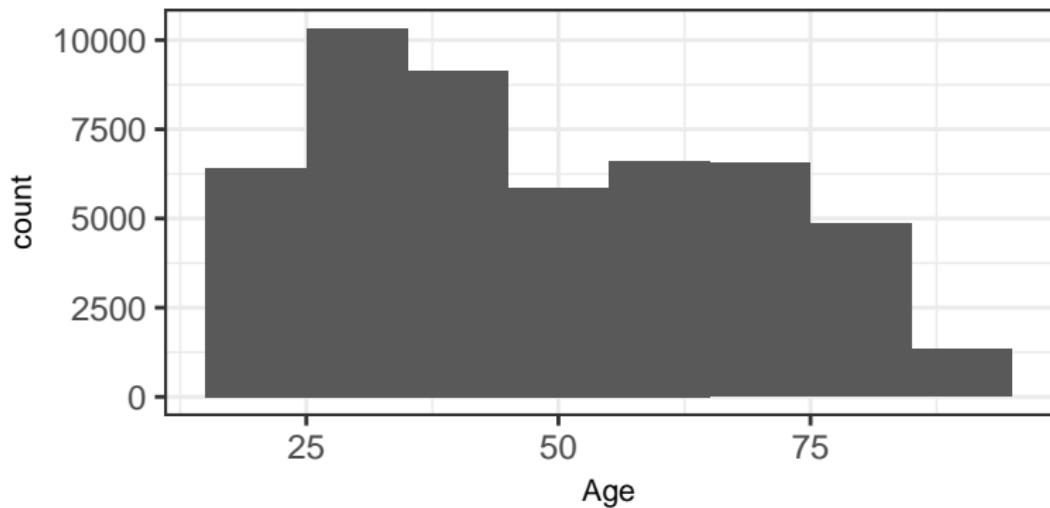
Histogram

```
ggplot(combined.long.df, aes(x = Age)) +  
  geom_histogram() + myTheme
```



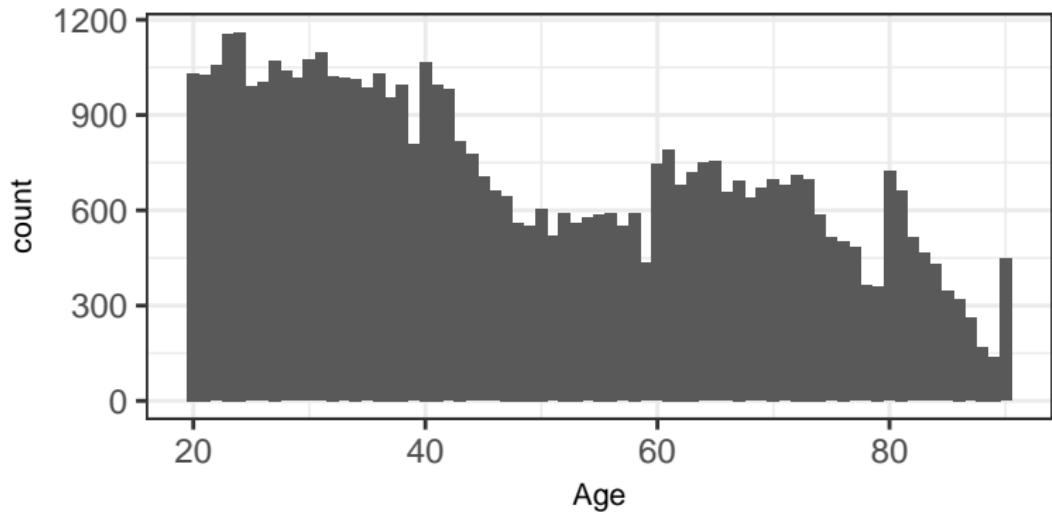
Histogram with wider binwidth

```
ggplot(combined.long.df, aes(x = Age)) +  
  geom_histogram(binwidth = 10) +  
  labs(x = "Age") + myTheme
```



Histogram with narrower binwidth

```
ggplot(combined.long.df, aes(x = Age)) +  
  geom_histogram(binwidth = 1) +  
  labs(x = "Age") + myTheme
```



Plot means in context

```
with(combined.long.df, tapply(Cholesterol, Age.group,  
                                mean, na.rm = TRUE))
```

```
## Under 35 36 to 60 Over 61  
## 186.2989 210.6702 221.1820
```

- Means are all but meaningless unless they are presented in context.
- Always present with standard deviations (SDs) or standard error of means (SEs) or confidence intervals.
- Plot means with 95% confidence intervals ($\pm 1.96 \times SE$).
 - $\pm 1 \times SE$ yields (approx.) a 68% confidence interval. Equivalent to using a 16% level of significance!!!!
 - $\pm 1 \times SD$ tells us **ABSOLUTELY NOTHING** about whether two means are statistically different from one another.

Calculating 95% CIs

- $95\% \text{ CI} = \text{Mean} \pm 1.96 \times \text{SE}$
- Standard Errors = $\frac{\text{Standard Deviation}}{\sqrt{\text{Sample Size}}}$

```
my.m <- with(combined.long.df, tapply(Cholesterol, Age.group,
                                         na.rm = TRUE))
```

```
my.m
```

```
## Under 35 36 to 60 Over 61
## 186.2989 210.6702 221.1820
```

```
my.sd <- with(combined.long.df, tapply(Cholesterol, Age.group,
                                         na.rm = TRUE))
```

```
my.sd
```

```
## Under 35 36 to 60 Over 61
## 38.98416 43.74297 45.62359
```

Calculating 95% CIs

```
my.n <- with(combined.long.df, tapply(Cholesterol, Age.group,
                                         function(x)length(which(!is.na(x)))))
```

```
my.n
```

```
## Under 35 36 to 60 Over 61
##      15780     17040     15366
```

```
my.stder <- my.sd/sqrt(my.n)
ci.upper <- my.m + 1.96*my.stder
ci.lower <- my.m - 1.96*my.stder
```

Calculating 95% CIs

```
my.stder <- my.sd/sqrt(my.n)
ci.upper <- my.m + 1.96*my.stder
ci.lower <- my.m - 1.96*my.stder

cbind(my.m, ci.lower, ci.upper)

##           my.m ci.lower ci.upper
## Under 35 186.2989 185.6907 186.9072
## 36 to 60 210.6702 210.0134 211.3270
## Over 61  221.1820 220.4606 221.9033
```

Calculating 95% CIs

```
y.df = data.frame(age.group = factor(names(my.m)),  
                   my.m, ci.upper, ci.lower)
```

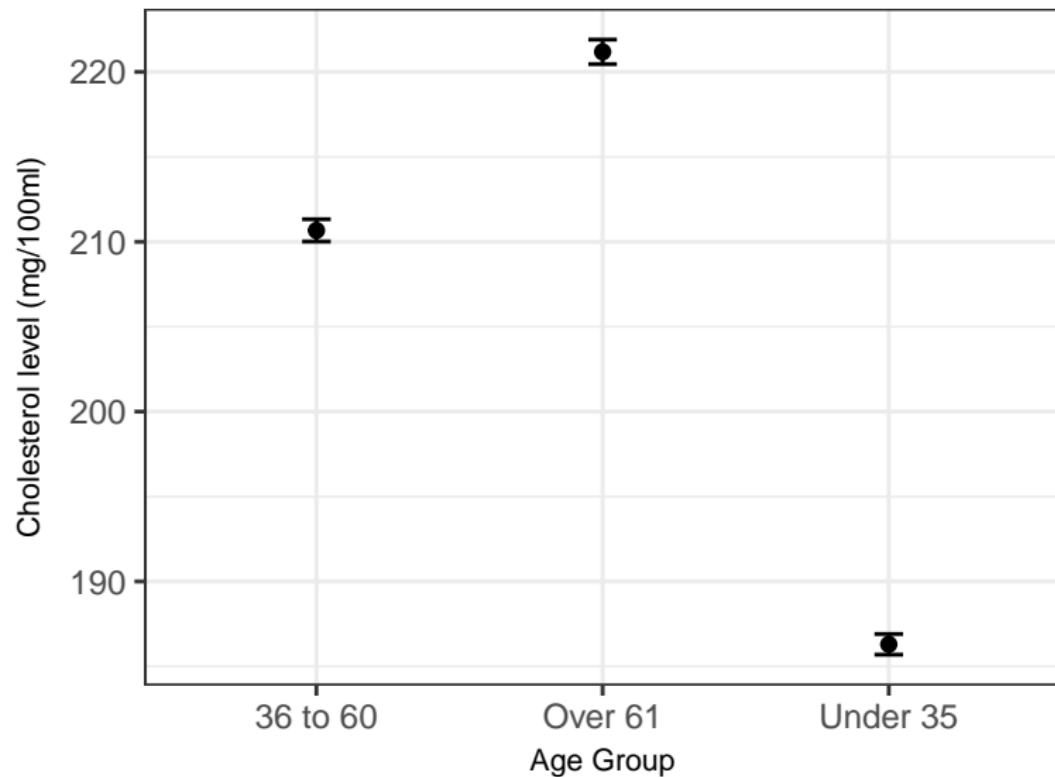
```
y.df
```

```
##           age.group      my.m ci.upper ci.lower  
## Under 35   Under 35 186.2989 186.9072 185.6907  
## 36 to 60   36 to 60 210.6702 211.3270 210.0134  
## Over 61    Over 61 221.1820 221.9033 220.4606
```

Errorbars

```
ggplot(y.df, aes(x = age.group, y = my.m)) + geom_point() +
  geom_errorbar(aes(ymax = ci.upper, ymin = ci.lower),
                width = 0.1) +
  xlab("Age Group") +
  ylab("Mean total feminist score") +
  myTheme
```

Errorbars



Any interaction between Gender and Age group?

```
GA.m <- with(combined.long.df, tapply(Cholesterol,
                                         list(Sex, Age.group), mean, na.rm = TRUE))
GA.m
```



```
##           Under 35 36 to 60  Over 61
## Female    184.7069 209.6103 231.4096
## Male      188.1431 211.8985 210.1405
```

Calculating 95% CIs

```
GA.sd <- with(combined.long.df,
               tapply(Cholesterol,
                      list(Sex, Age.group),
                      sd, na.rm = TRUE))

GA.n <- with(combined.long.df, tapply(Cholesterol,
                                         list(Sex, Age.group),
                                         function(x)length(which(!is.na(x)))))

GA.stder <- GA.sd/sqrt(GA.n)
GA.upper <- GA.m + 1.96*GA.stder
GA.lower <- GA.m - 1.96*GA.stder
```

Calculating 95% CIs

```
GA.df <- data.frame(  
  Age.group = factor(rep(colnames(GA.m), 2),  
                      levels = colnames(GA.m)),  
  Gender = rep(rownames(GA.m), c(3, 3)),  
  Mean = c(GA.m[1,], GA.m[2,]),  
  Upper = c(GA.upper[1,], GA.upper[2,]),  
  Lower = c(GA.lower[1,], GA.lower[2,]))  
)
```

Calculating 95% CIs

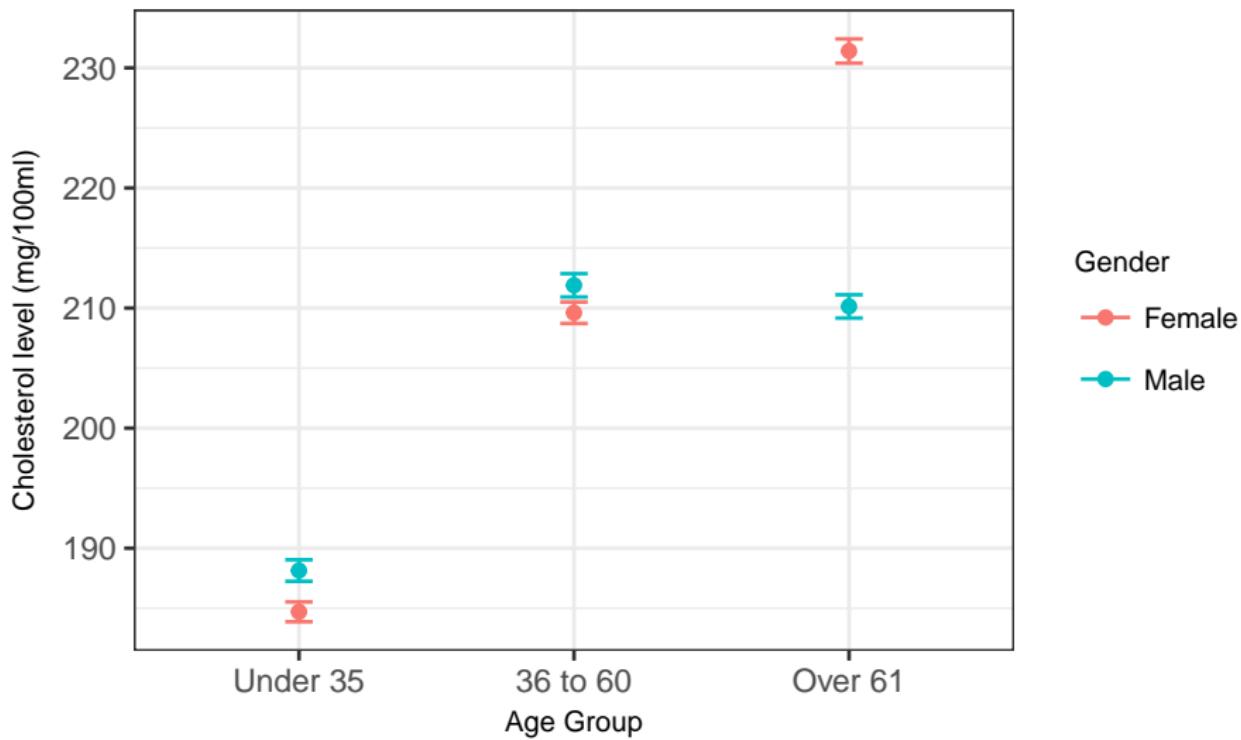
```
##   Age.group Gender      Mean      Upper      Lower
## 1 Under 35 Female 184.7069 185.5337 183.8801
## 2 36 to 60 Female 209.6103 210.5015 208.7191
## 3 Over 61 Female 231.4096 232.4182 230.4009
## 4 Under 35 Male 188.1431 189.0392 187.2469
## 5 36 to 60 Male 211.8985 212.8694 210.9276
## 6 Over 61 Male 210.1405 211.1114 209.1695
```

Plotting mean \pm 95% CI:

```
g <- ggplot(GA.df, aes(x = Age.group, y = Mean,
                        color = Gender)) +
  xlab("Age Group") +
  ylab("Cholesterol level (mg/100ml)")

g + geom_point() +
  geom_errorbar(aes(ymax = Upper, ymin = Lower),
                width = 0.1) +
  myTheme
```

Plotting mean \pm 95% CI:

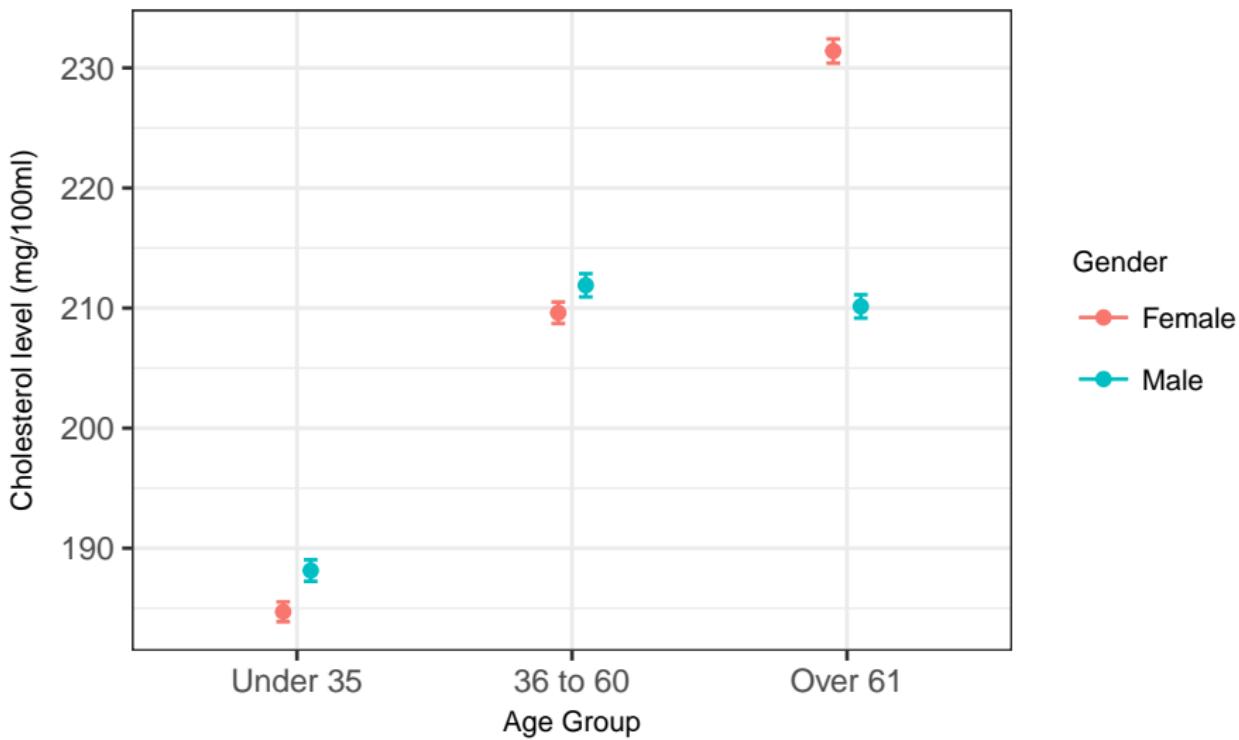


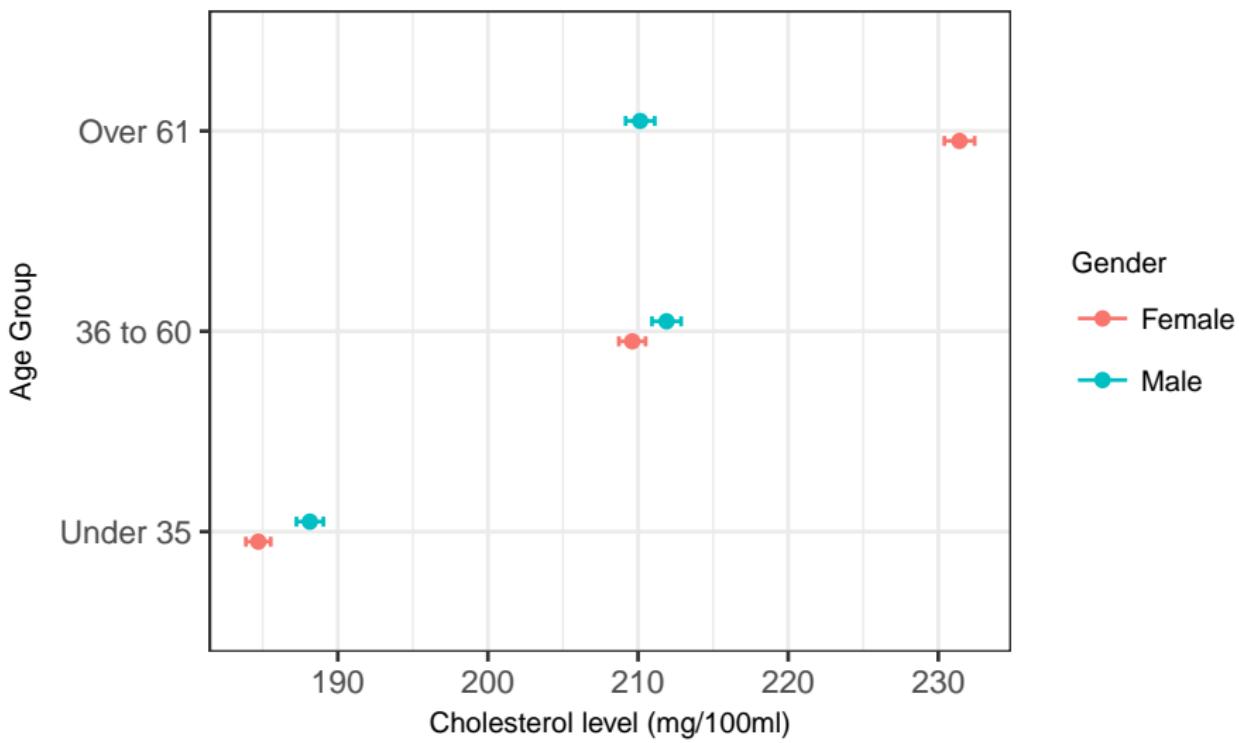
Side-by-side?

```
dodge <- position_dodge(width=0.2)

g + geom_point(position = dodge) +
  geom_errorbar(aes(ymax = Upper, ymin = Lower),
                width = 0.1, position = dodge) +
  myTheme
```

Side-by-side

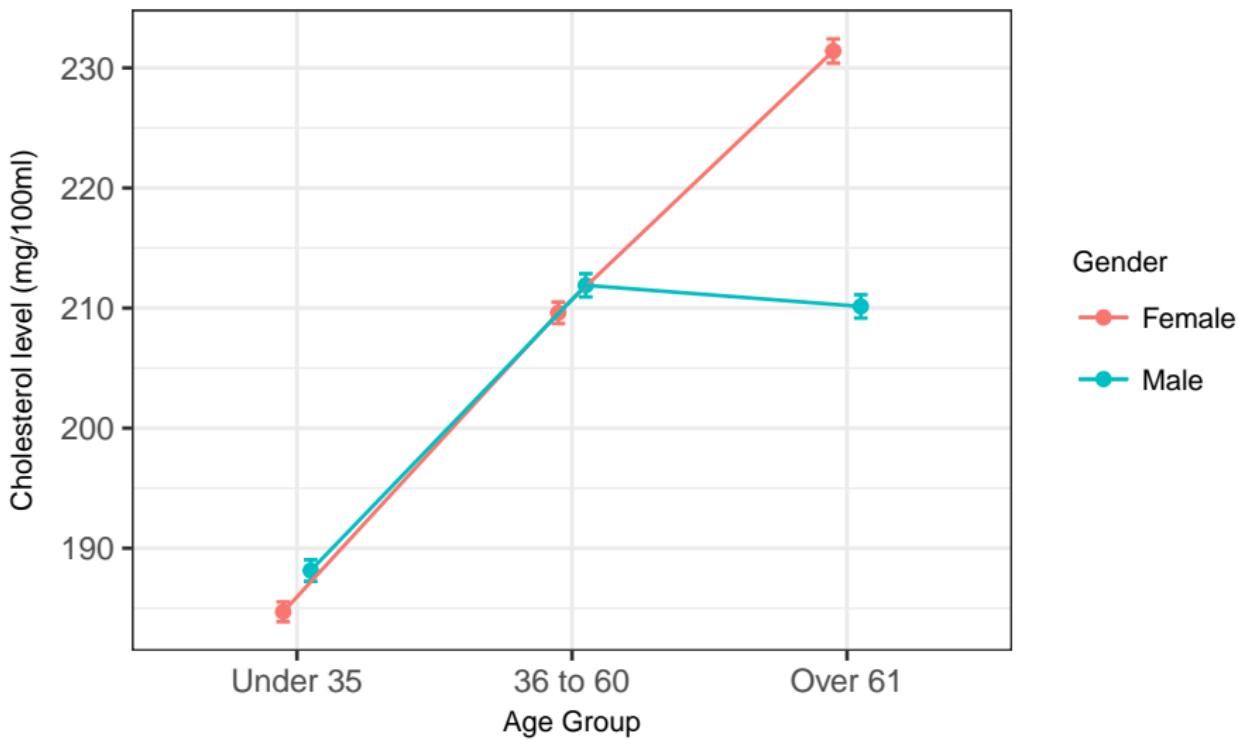




Connect the points?

```
g + geom_point(position = dodge) +
  geom_errorbar(aes(ymax = Upper, ymin = Lower),
                width = 0.1, position = dodge) +
  geom_path(aes(x = as.numeric(Age.group)),
            position = dodge) +
myTheme
```

Connect the points



Save a ggplot

```
ggsave("mtcars.pdf")
ggsave("mtcars.png")

ggsave("mtcars.pdf", width = 4, height = 4)
ggsave("mtcars.pdf", width = 20, height = 20, units = "cm")
```

Summary

Plot Types	geom functions
Scatterplot	geom_point()
Bars chart	geom_bar()
Histogram	geom_histogram()
Boxplot	geom_boxplot()
Line plot	geom_path()
Errorbar	geom_errorbar()

- ggplot2 Documentation: <http://docs.ggplot2.org/current/>
- cheatsheets:
<https://www.rstudio.com/resources/cheatsheets/>