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

2 March, 2017

THE UNIVERSITY OF AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND

SCIENCE
DEPARTMENT OF STATISTICS

ggplot2 package

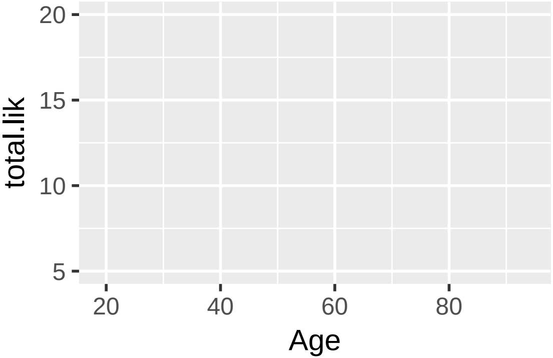
- Documentation: http://docs.ggplot2.org/current/
- recommended reading “The Layered Grammar of Graphics”: http://vita.had.co.nz/papers/layered-grammar.pdf
- Load ggplot2 package

```
library(ggplot2)
```

Create a new ggplot

- Initialising a ggplot object.

```
ggplot(data = issp.df,
       mapping = aes(x = Age, y = total.lik))
```



Create a new ggplot

- Initialising a ggplot object.

```
ggplot(data = issp.df,
       mapping = aes(x = Age, y = total.lik))
```

There are three common ways to invoke ggplot:

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

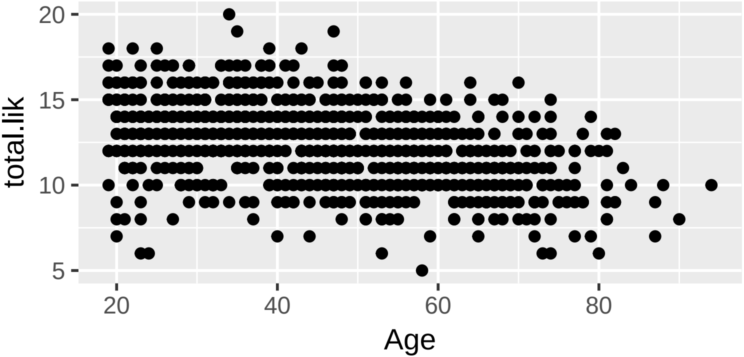
Create a new ggplot

- assign this ggplot object to a variable

```
g <- ggplot(data = issp.df,
       mapping = aes(x = Age, y =  total.lik))
```

Create a Scatterplot

```r
g + geom_point()
```



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

Create a Scatterplot

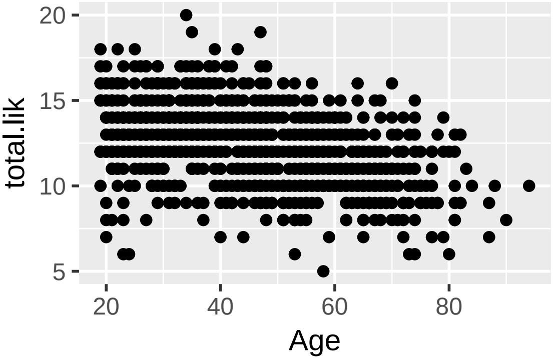
- Note that here are three common ways to invoke ggplot:

```
ggplot(data = issp.df,
       mapping = aes(x = Age, y =  total.lik)) + geom_point()
ggplot(data = issp.df) +
  geom_point( mapping = aes(x = Age, y =  total.lik))
ggplot() +
  geom_point(data = issp.df,
             mapping = aes(x = Age, y =  total.lik))
```

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

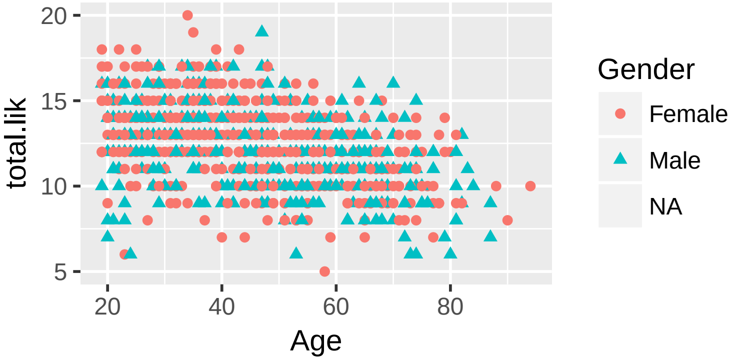
First method is recommended

```
g <- ggplot(data = issp.df,
        mapping = aes(x = Age, y =  total.lik))
g + geom_point()
```



Control colour and shape

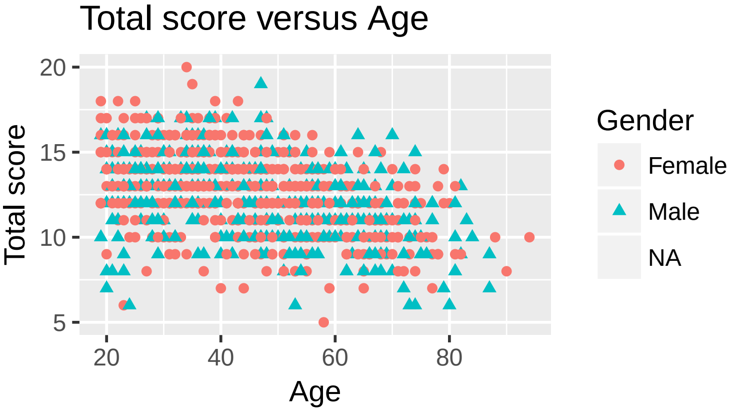
```
g + geom_point(aes(colour = Gender, shape = Gender))
```



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

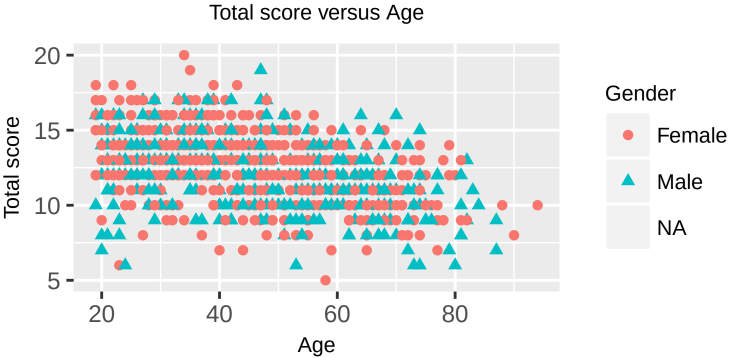
Modify axis, legend, and plot labels

```
(g <- g + geom_point(aes(colour = Gender, shape = Gender)) +
   labs(title = "Total score versus Age",
   x = "Age", y = "Total score"))
```



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

```r
g <- ggplot(
  data = na.omit(issp.df[,c("Age", "total.lik", "Gender")]),
  mapping = aes(x = Age, y =  total.lik,
                color = Gender, shape = Gender)) +
  geom_point() +
  labs(title ="Total score versus Age",
        x = "Age", y = "Total score")
```

- ggplot object can be further modified.

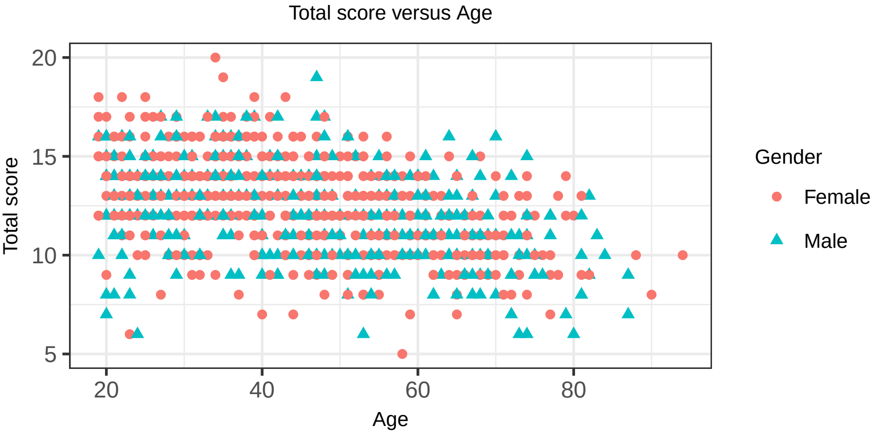
Create a Scatterplot

```
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.

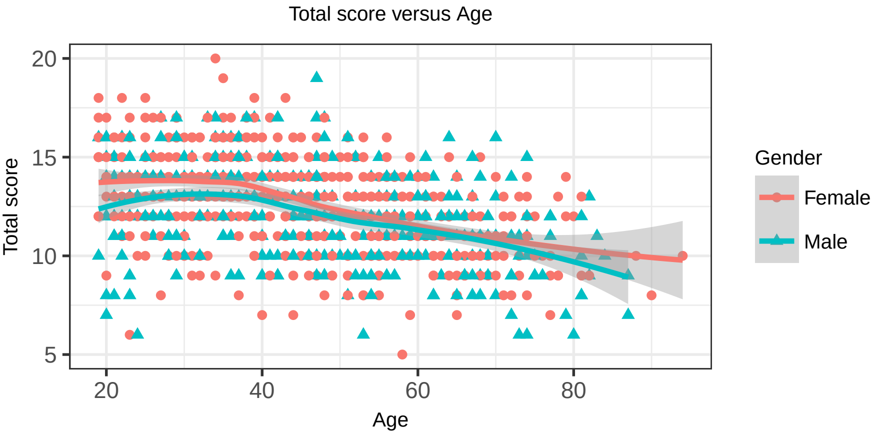
Scatterplot

```
g +  myTheme
```



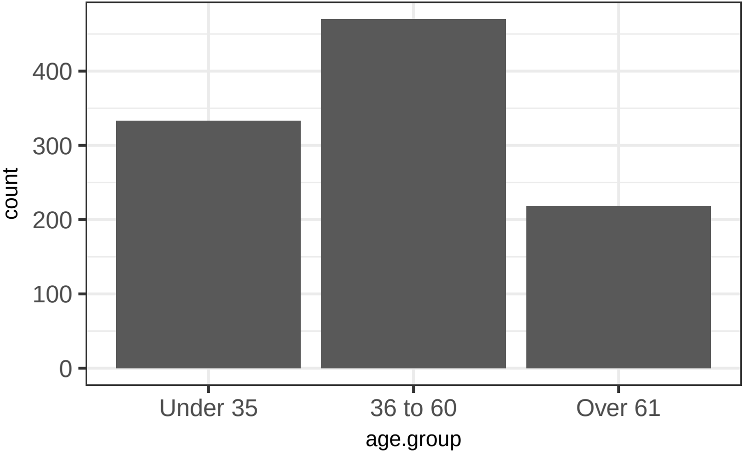
Scatterplot with a smoother

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



Bar chart

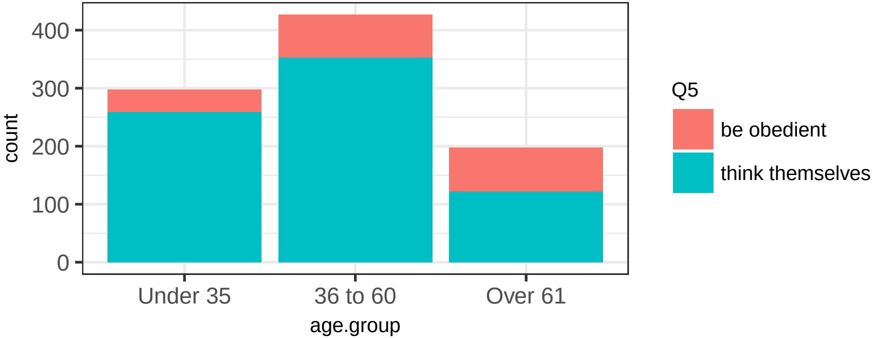
```
ggplot(na.omit(issp.df[,c("ID", "age.group")]),
       aes(x = age.group)) + geom_bar() + myTheme
```



Importance of childhood obedience by age group

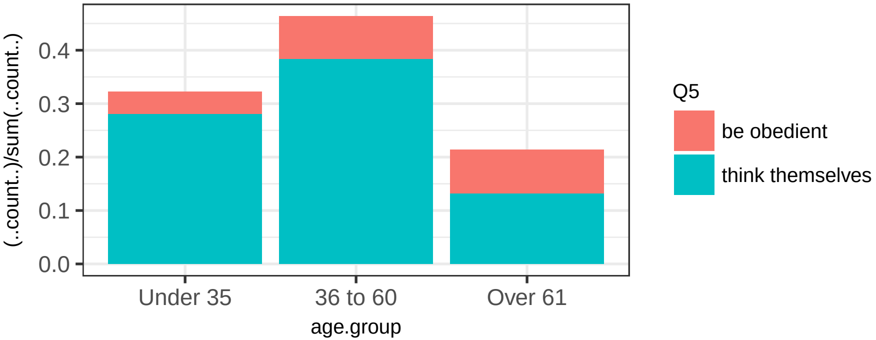
Q5: Is obedience important in terms of preparing children for life?

```
ggplot(na.omit(issp.df[,c("ID", "age.group", "Q5")]),
       aes(x = age.group, fill = Q5)) +
      geom_bar() + myTheme
```



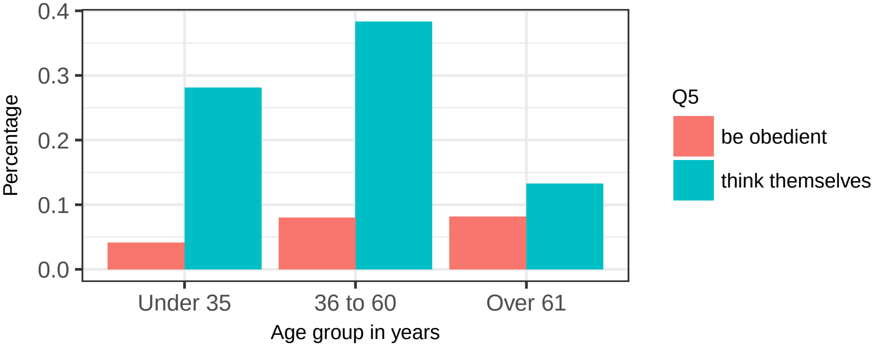
Bar chart on proportions

```
g <- ggplot(na.omit(issp.df[,c("ID", "age.group", "Q5")]),
  aes(x = age.group, y = (..count..)/sum(..count..),
      fill = Q5))
g + geom_bar() + myTheme
```



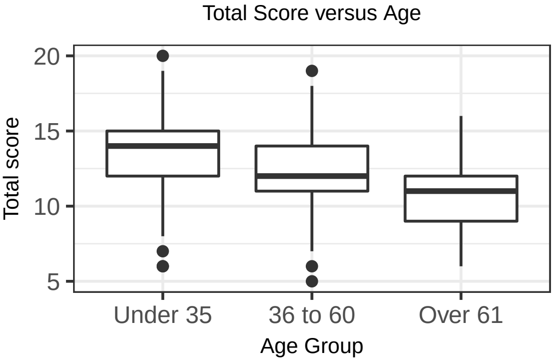
Bar chart on proportions

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



Boxplot using geom_boxplot()

```
ggplot(na.omit(issp.df[,c("age.group", "total.lik")]),
       aes(x = age.group, y = total.lik)) +
  geom_boxplot()  +
  labs(title = "Total Score versus Age",
      x = "Age Group", y = "Total score") +  myTheme
```

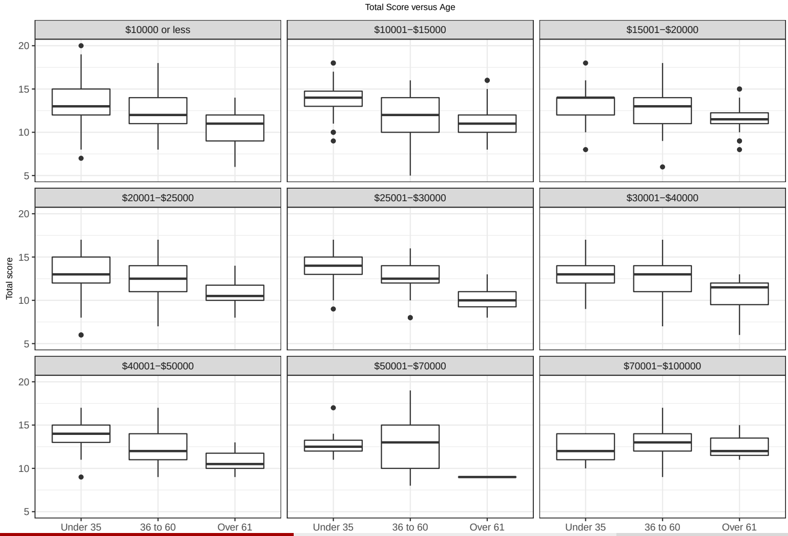


Boxplot with panels using facet_wrap()

```
g <- ggplot(na.omit(issp.df[,c("ID", "age.group",
                               "total.lik", "Income")]),
       aes(x = age.group, y = total.lik)) +
  geom_boxplot()  + labs(title = "Total Score versus Age",
            x = "Age Group", y = "Total score")
```

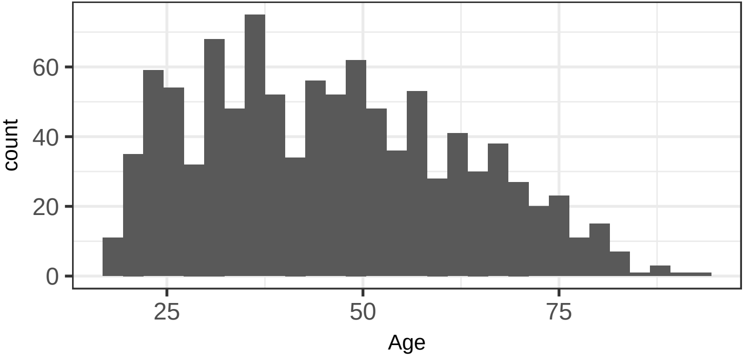
Boxplot with panels using facet_wrap()

```
g + facet_wrap(~Income) + myTheme
```



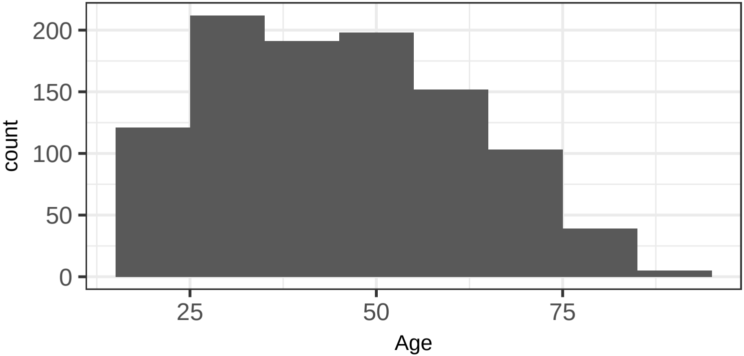
Histogram

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



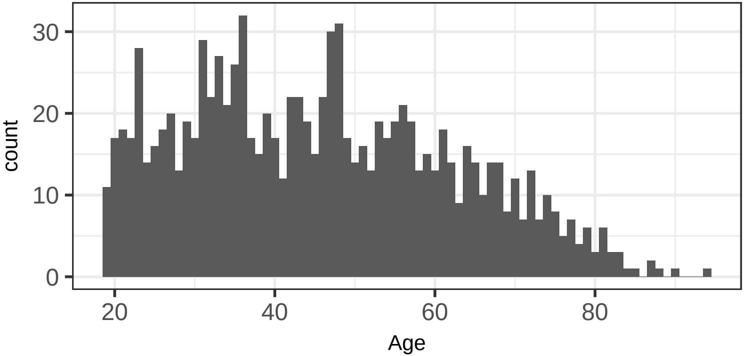
Histogram with wider binwidth

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



Histogram with narrower binwidth

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



Plot means in context

```r
with(issp.df, tapply(total.lik, age.group,
                     mean, na.rm = TRUE))
```

```
## Under 35 36 to 60  Over 61
## 13.38871 12.45516 10.78836
```

- 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 (± 1.96 × SE).
  - ± 1 × SE yields (approx.) a 68% confidence interval. Equivalent to using a 16% level of significance!!!!
  - ± 1 × SD tells us **ABSOLUTELY NOTHING** about whether two means are statistically different from one another.

Calculating 95% CIs

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

```
my.m <- with(issp.df, tapply(total.lik, age.group, mean,
                             na.rm = TRUE))
my.m
```

```
## Under 35 36 to 60  Over 61
## 13.38871 12.45516 10.78836
```

```
my.sd <- with(issp.df, tapply(total.lik, age.group, sd,
                              na.rm = TRUE))
my.sd
```

```
## Under 35 36 to 60  Over 61
## 2.139623 2.156049 1.964491
```

Calculating 95% CIs

```
my.n <- with(issp.df, tapply(total.lik, age.group,
            function(x)length(which(!is.na(x)))))
my.n
```

```
## Under 35 36 to 60  Over 61
##      319      446      189
```

```
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

```r
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 13.38871 13.15391 13.62351
## 36 to 60 12.45516 12.25506 12.65526
## Over 61  10.78836 10.50828 11.06844
```

Calculating 95% CIs

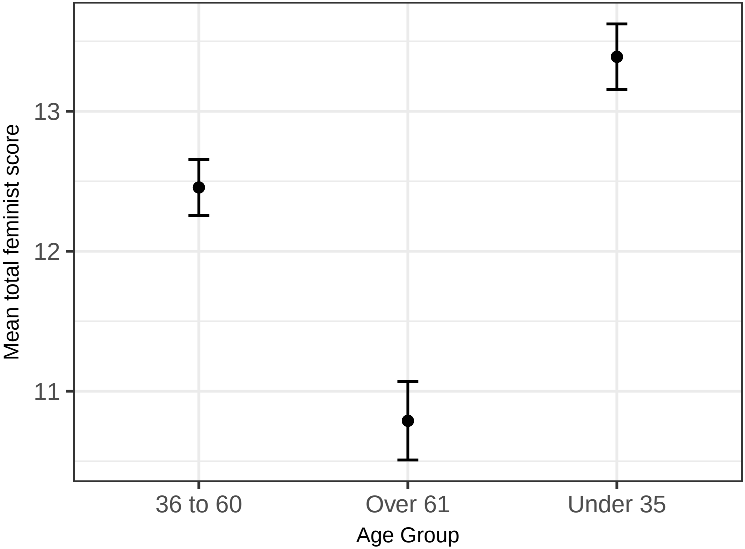
```
y.df = data.frame(age.group =  factor(names(my.m)),
                  my.m, ci.upper, ci.lower)

y.df
```

```
##           age.group     my.m ci.upper ci.lower
## Under 35  Under 35 13.38871 13.62351 13.15391
## 36 to 60  36 to 60 12.45516 12.65526 12.25506
## Over 61    Over 61 10.78836 11.06844 10.50828
```

Errorbars

```r
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


Mean total feminist score
13
12
11
36 to 60
Over 61
Under 35
Age Group

Any interaction between Gender and Age group?

```r
GA.m <- with(issp.df, tapply(total.lik,
           list(Gender, age.group), mean, na.rm = TRUE))
GA.m
```

```
##        Under 35 36 to 60  Over 61
## Female  13.7500 12.64576 10.87234
## Male    12.8963 12.16092 10.70526
```

Calculating 95% CIs

```
GA.sd <- with(issp.df,
              tapply(total.lik,
                     list(Gender, age.group),
                          sd, na.rm = TRUE))

GA.n <- with(issp.df, tapply(total.lik,
      list(Gender,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,])
  )
```

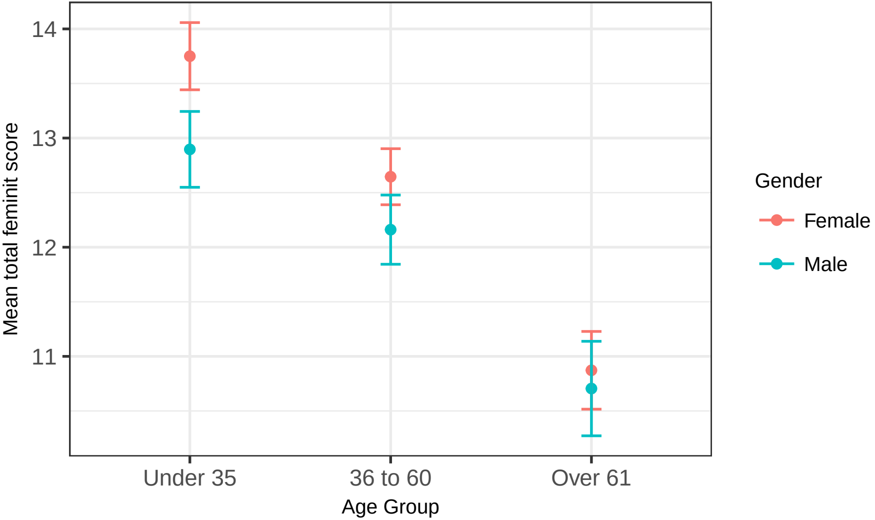
Calculating 95% CIs

```
##   Age.group Gender     Mean    Upper    Lower
## 1  Under 35 Female 13.75000 14.05782 13.44218
## 2  36 to 60 Female 12.64576 12.90255 12.38896
## 3   Over 61 Female 10.87234 11.22838 10.51630
## 4  Under 35   Male 12.89630 13.24377 12.54882
## 5  36 to 60   Male 12.16092 12.47794 11.84390
## 6   Over 61   Male 10.70526 11.13822 10.27230
```

Plotting mean ± 95% CI:

```r
g <- ggplot(GA.df, aes(x = Age.group, y = Mean,
                 color = Gender)) +
  xlab("Age Group")+
  ylab("Mean total feminit score")

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

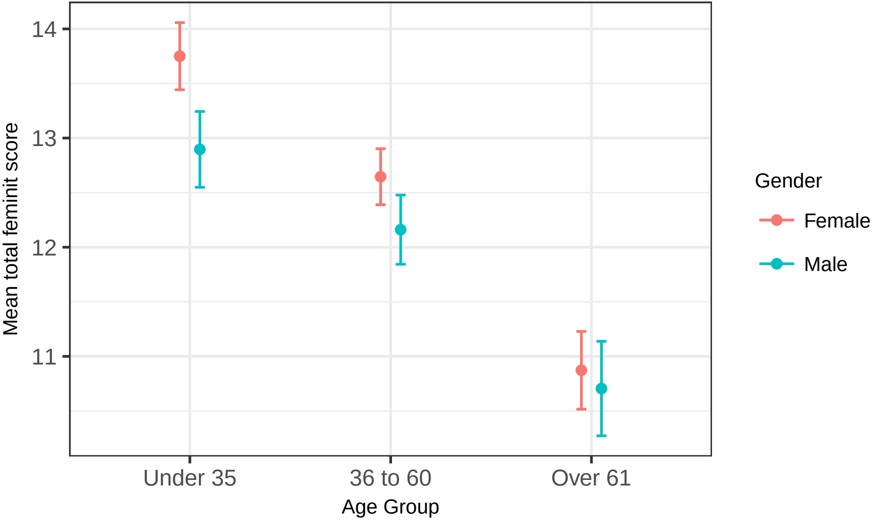
Plotting mean ± 95% CI:



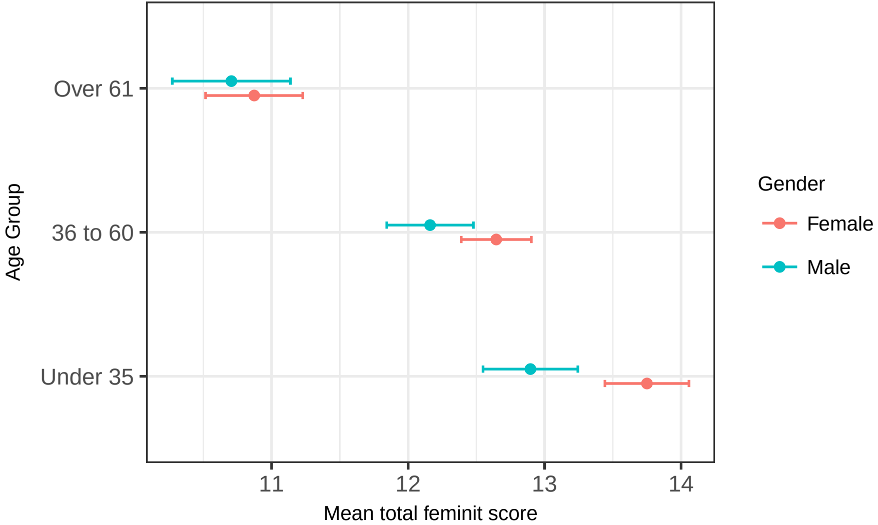
Side-by-side?

```r
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


14
13
12
11
Mean total feminit score
Under 35
36 to 60
Over 61
Age Group
Gender
Female
Male

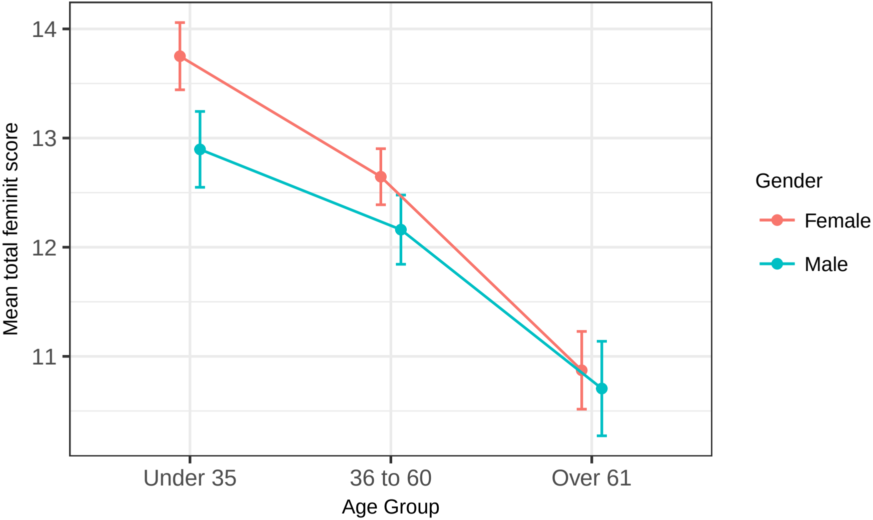
Flip


Over 61
36 to 60
Under 35
Age Group
11
12
13
14
Mean total feminit score
Gender
Female
Male

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

```r
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/`