HDAT9800 Health Data Visualisation & Communication

Chapter 4 Interactive Tutorial -- more on visualisation with `ggplot2`

Tim Churches

UNSW Medicine

22nd June 2022

Agenda for Chapter 4 ggplot2 interactive session

- recap core readings (Wilke Chapters 6)
- reproduce and extend the example charts Wilke provides using different (medical) datasets
- a quick look at the `patchwork` package

Wilke Chapter 6 - visualising amounts

- let's use the `medicaldata` library again
- if not installed:

```
remotes::install_github("CBDRH-HDAT9800/medicaldata")
```

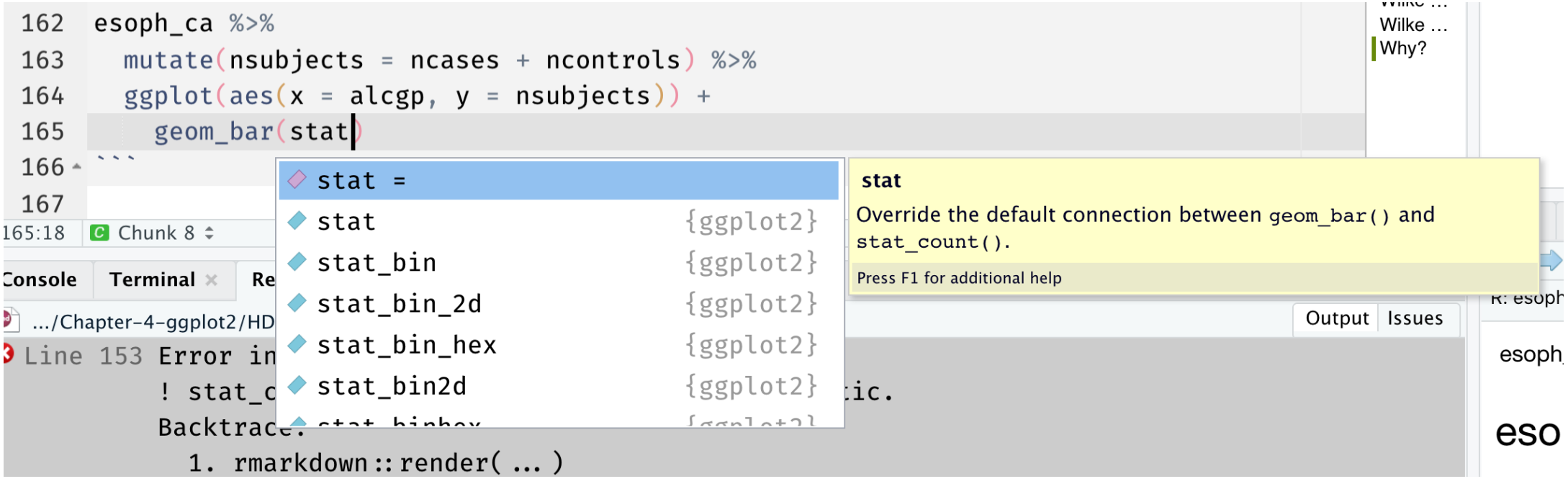
Wilke section 6.1 - Bar plots

- write some R code using `ggplot2` to make a bar chart using the `esoph_ca` dataset in the `medicaldata` package to create a bar chart which shows:
  - `nsubjects` on the y-axis (you'll need to add (sum) `ncases` and `ncontrols` to create an `nsubjects` column using `dplyr`
  - `alcgrp` on the x-axis
- examine the `esoph_ca` dataset and its documentation (`?esoph_ca`) first!

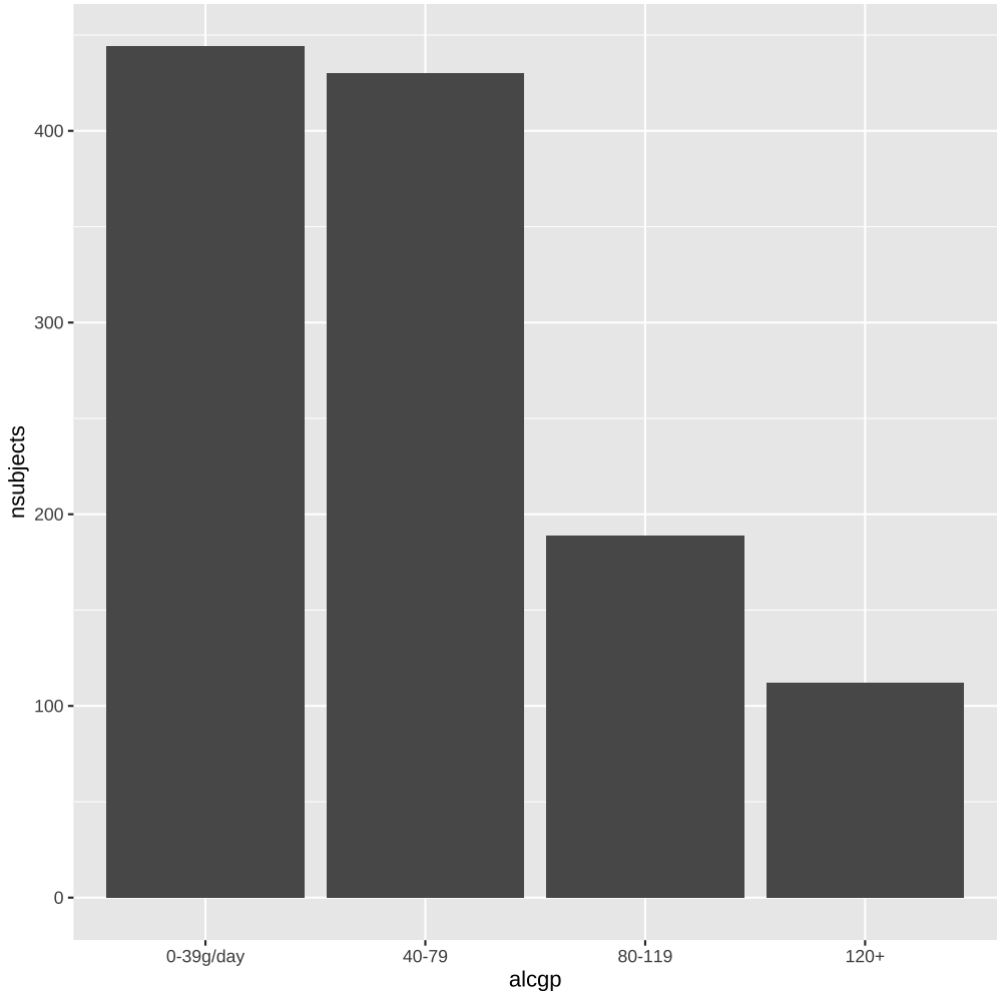
```r
library(tidyverse)
library(medicaldata)

esoph_ca %>%
  mutate(nsubjects = ncases + ncontrols) %>%
  ggplot(aes(x = alcgp, y = nsubjects)) +
    geom_bar()
```

Why?

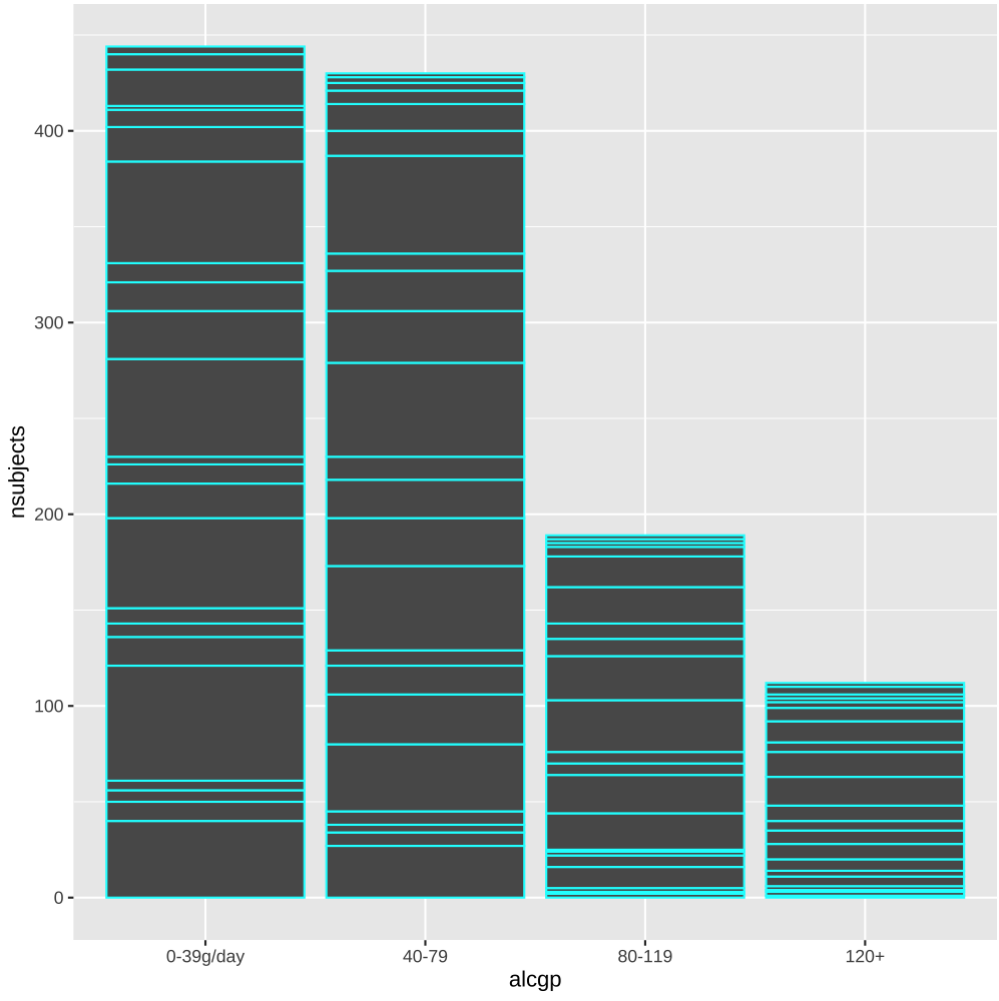
```
Line 153 Error in `f()`:
         ! stat_count() can only have an x or y aesthetic.
         Backtrace:
           1. rmarkdown::render( ... )
           2. knitr::knit(knit_input, knit_output, envir = envir, quiet = quiet)
           3. knitr:::process_file(text, output)
           6. knitr:::process_group.block(group)
           7. knitr:::call_block(x)
                ...
          29. ggplot2 f(l = layers[[i]], d = data[[i]])
          30. l$compute_statistic(d, layout)
          31. ggplot2 f( ... , self = self)
          32. self$stat$setup_params(data, self$stat_params)
          33. ggplot2 f( ... )
         Execution halted
```


esoph_ca %>%
mutate(nsubjects = ncases + ncontrols) %>%
ggplot(aes(x = alcgp, y = nsubjects)) +
geom_bar(stat
```
stat =
stat {ggplot2}
stat_bin {ggplot2}
stat_bin_2d {ggplot2}
stat_bin_hex {ggplot2}
stat_bin2d {ggplot2}
stat
Override the default connection between geom_bar() and stat_count().
Press F1 for additional help
165:18
Chunk 8
Console
Terminal
Output
Issues
.../Chapter-4-ggplot2/HD
Line 153 Error in
! stat_c
Backtrace:
1. rmarkdown::render( ... )
Wilke ...
Why?
esoph
eso
```

```
esoph_ca %>%
  mutate(nsubjects = ncases + ncontrols) %>%
  ggplot(aes(x = alcgp, y = nsubjects)) +
    geom_bar(stat = "identity")
```


400
300
200
100
0
nsubjects
0-39g/day
40-79
80-119
120+
alcgp

```
esoph_ca %>%
  mutate(nsubjects = ncases + ncontrols) %>%
  ggplot(aes(x = alcgp, y = nsubjects)) +
    geom_bar(stat = "identity", col = "cyan")
```


nsubjects
400
300
200
100
0
0-39g/day
40-79
80-119
120+
alcgp

```
esoph_ca %>%
  mutate(nsubjects = ncases + ncontrols)
```

```
##    agegp     alcgp    tobgp ncases ncontrols nsubjects
## 1  25-34 0-39g/day 0-9g/day      0        40         40
## 2  25-34 0-39g/day    10-19      0        10         10
## 3  25-34 0-39g/day    20-29      0         6          6
## 4  25-34 0-39g/day      30+      0         5          5
## 5  25-34     40-79 0-9g/day      0        27         27
## 6  25-34     40-79    10-19      0         7          7
## 7  25-34     40-79    20-29      0         4          4
## 8  25-34     40-79      30+      0         7          7
## 9  25-34    80-119 0-9g/day      0         2          2
## 10 25-34    80-119    10-19      0         1          1
## 11 25-34    80-119      30+      0         2          2
## 12 25-34      120+ 0-9g/day      0         1          1
## 13 25-34      120+    10-19      1         1          2
## 14 25-34      120+    20-29      0         1          1
## 15 25-34      120+      30+      0         2          2
## 16 35-44 0-39g/day 0-9g/day      0        60         60
## 17 35-44 0-39g/day    10-19      1        14         15
## 18 35-44 0-39g/day    20-29      0         7          7
## 19 35-44 0-39g/day      30+      0         8          8
## 20 35-44     40-79 0-9g/day      0        35         35
## 21 35-44     40-79    10-19      3        23         26
```

```
esoph_ca %>%
  mutate(nsubjects = ncases + ncontrols) %>%
  group_by(alcgp) %>%
  summarise(nsubjects = sum(nsubjects))
```

```
## # A tibble: 4 × 2
##   alcgp     nsubjects
##   <ord>         <dbl>
## 1 0-39g/day       444
## 2 40-79           430
## 3 80-119          189
## 4 120+            112
```

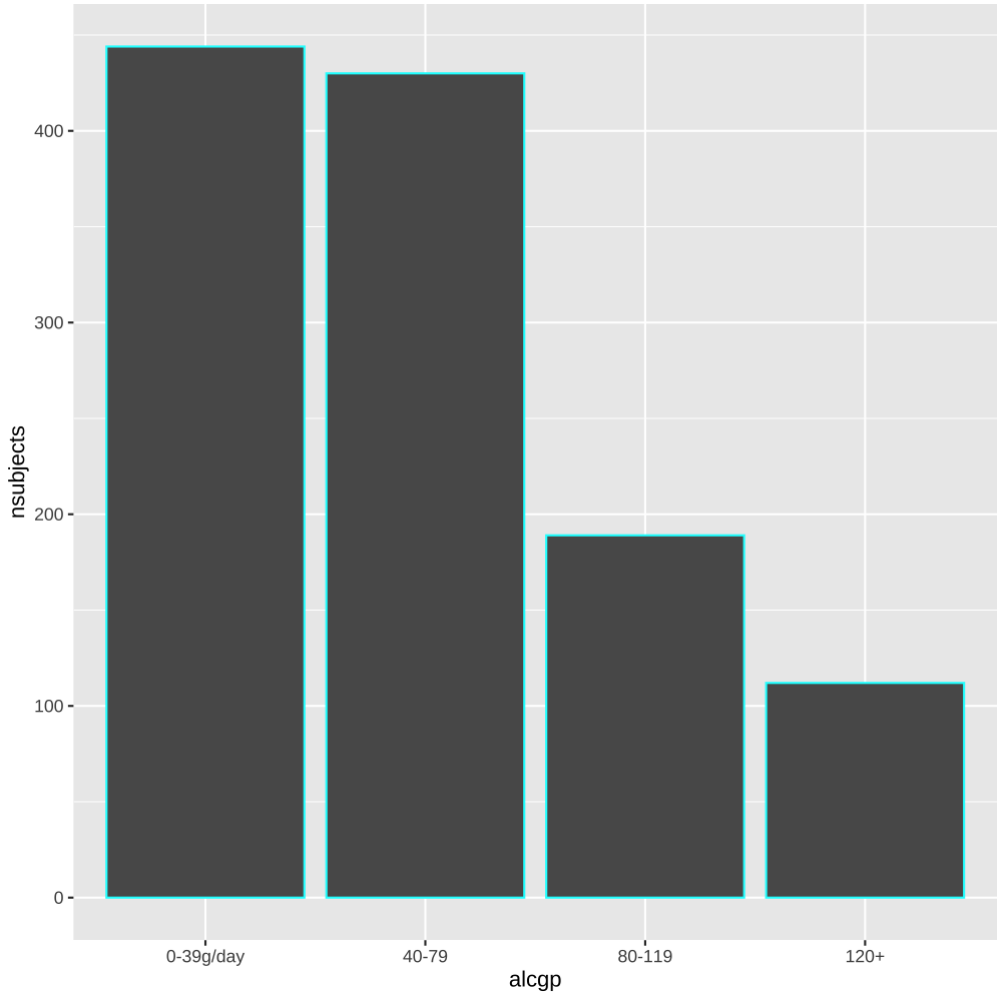
```
esoph_ca %>%
  group_by(alcgp) %>%
  summarise(nsubjects = sum(ncases) + sum(ncontrols))
```

```
## # A tibble: 4 × 2
##   alcgp     nsubjects
##   <ord>         <dbl>
## 1 0-39g/day       444
## 2 40-79           430
## 3 80-119          189
## 4 120+            112
```

```
esoph_ca %>%
  group_by(alcgp) %>%
  summarise(nsubjects = sum(ncases + ncontrols))
```

```
## # A tibble: 4 × 2
##   alcgp     nsubjects
##   <ord>         <dbl>
## 1 0-39g/day       444
## 2 40-79           430
## 3 80-119          189
## 4 120+            112
```

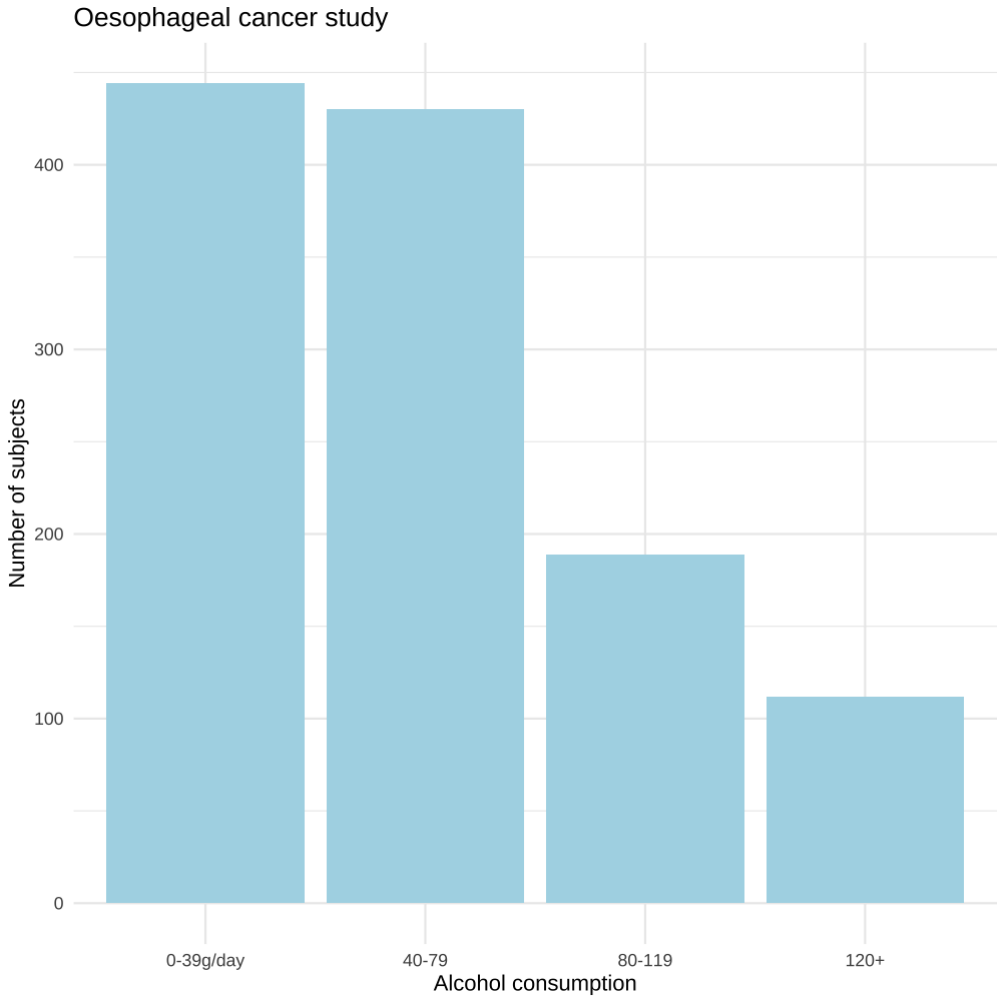
```
esoph_ca %>%
  group_by(alcgp) %>%
  summarise(nsubjects = sum(ncases + ncontrols)) %>%
  ggplot(aes(x = alcgp, y = nsubjects)) +
    geom_bar(stat = "identity", colour = "cyan")
```


nsubjects
400
300
200
100
0
0-39g/day
40-79
80-119
120+
alcgp

Make it prettier

- add `theme_minimal()`
- choose and set a fill colour for the bars
- label the x-axis and y-axis better

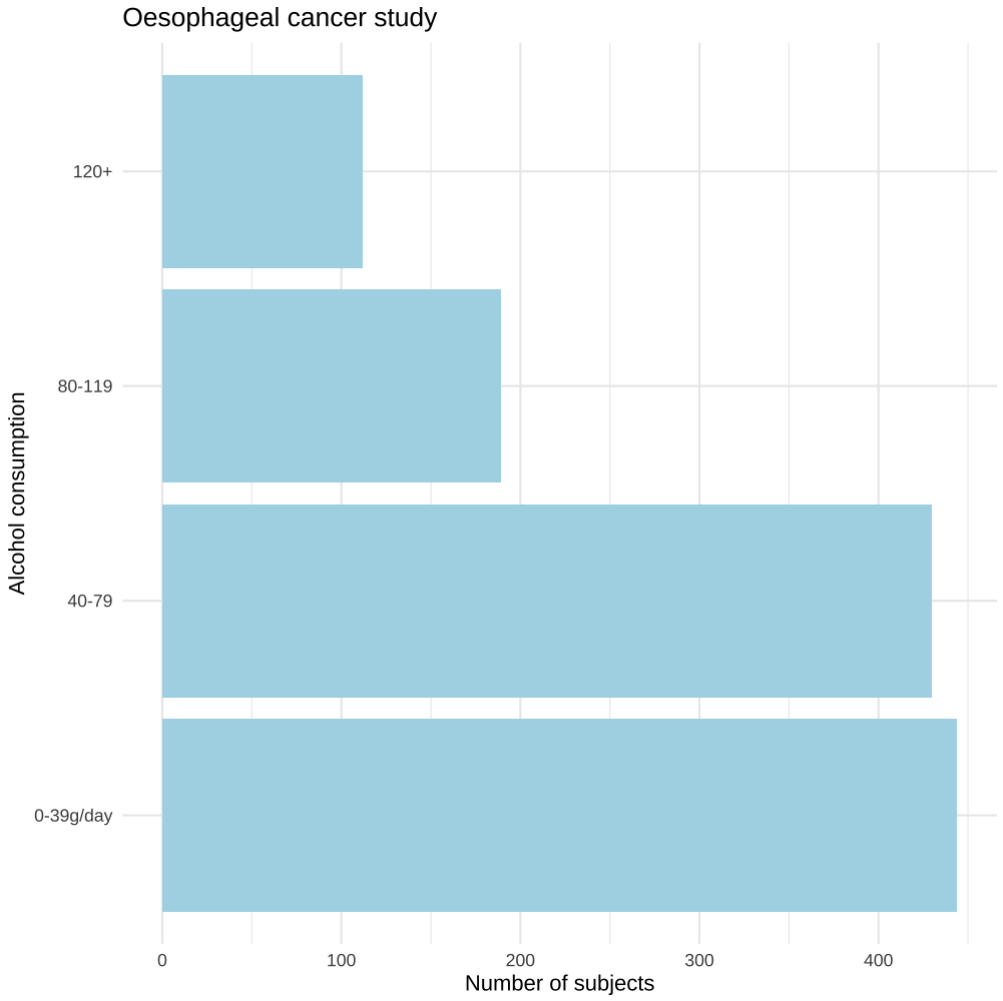
```
esoph_ca %>%
  group_by(alcgp) %>%
  summarise(nsubjects = sum(ncases + ncontrols)) %>%
  ggplot(aes(x = alcgp, y = nsubjects)) +
    geom_bar(stat = "identity", fill = "lightblue") +
    labs(x="Alcohol consumption",
         y="Number of subjects",
         title="Oesophageal cancer study") +
    theme_minimal()
```


Oesophageal cancer study
Number of subjects
400
300
200
100
0
0-39g/day
40-79
80-119
120+
Alcohol consumption

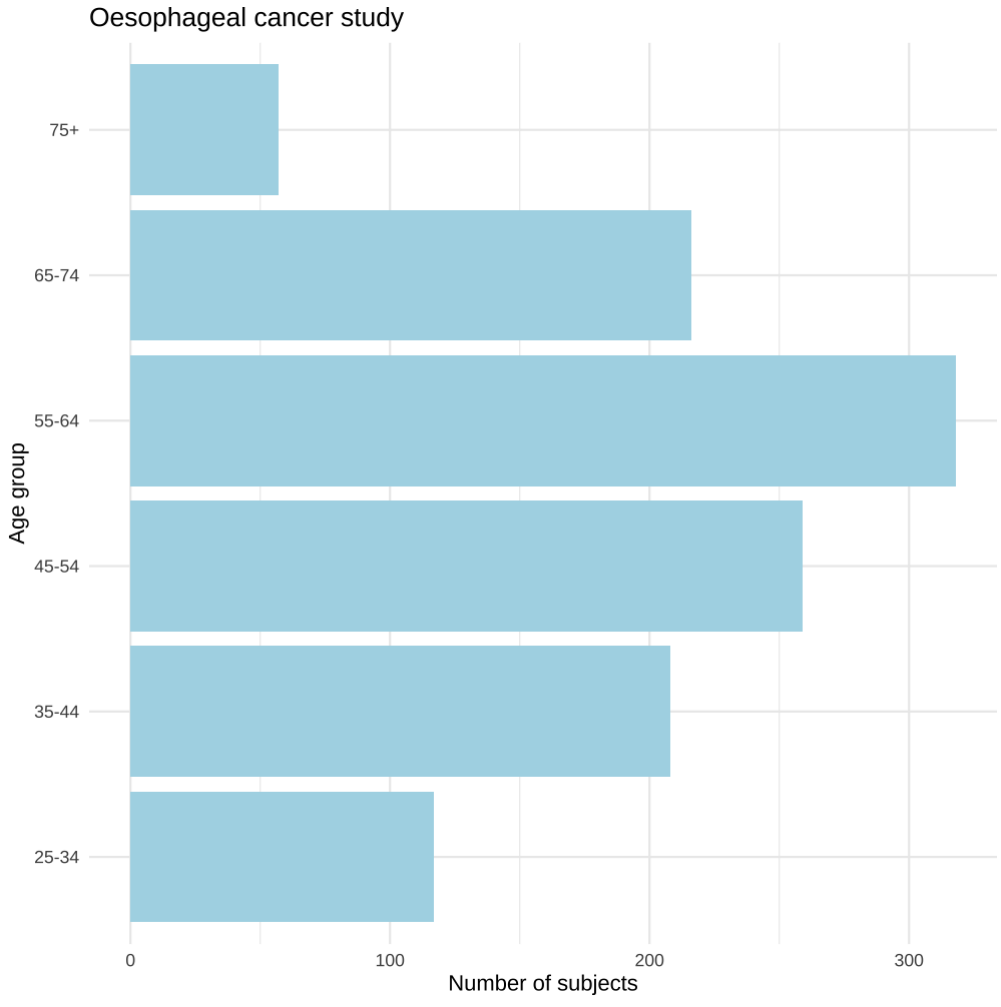
Return to Wilke section 6.1

Flip the coordinates

```
p <- esoph_ca %>%
  group_by(alcgp) %>%
  summarise(nsubjects = sum(ncases + ncontrols)) %>%
  ggplot(aes(x = alcgp, y = nsubjects)) +
    geom_bar(stat = "identity", fill = "lightblue") +
    labs(x="Alcohol consumption",
         y="Number of subjects",
         title="Oesophageal cancer study") +
    theme_minimal()

p + coord_flip()
```


Oesophageal cancer study
Alcohol consumption
120+
80-119
40-79
0-39g/day
0
100
200
300
400
Number of subjects

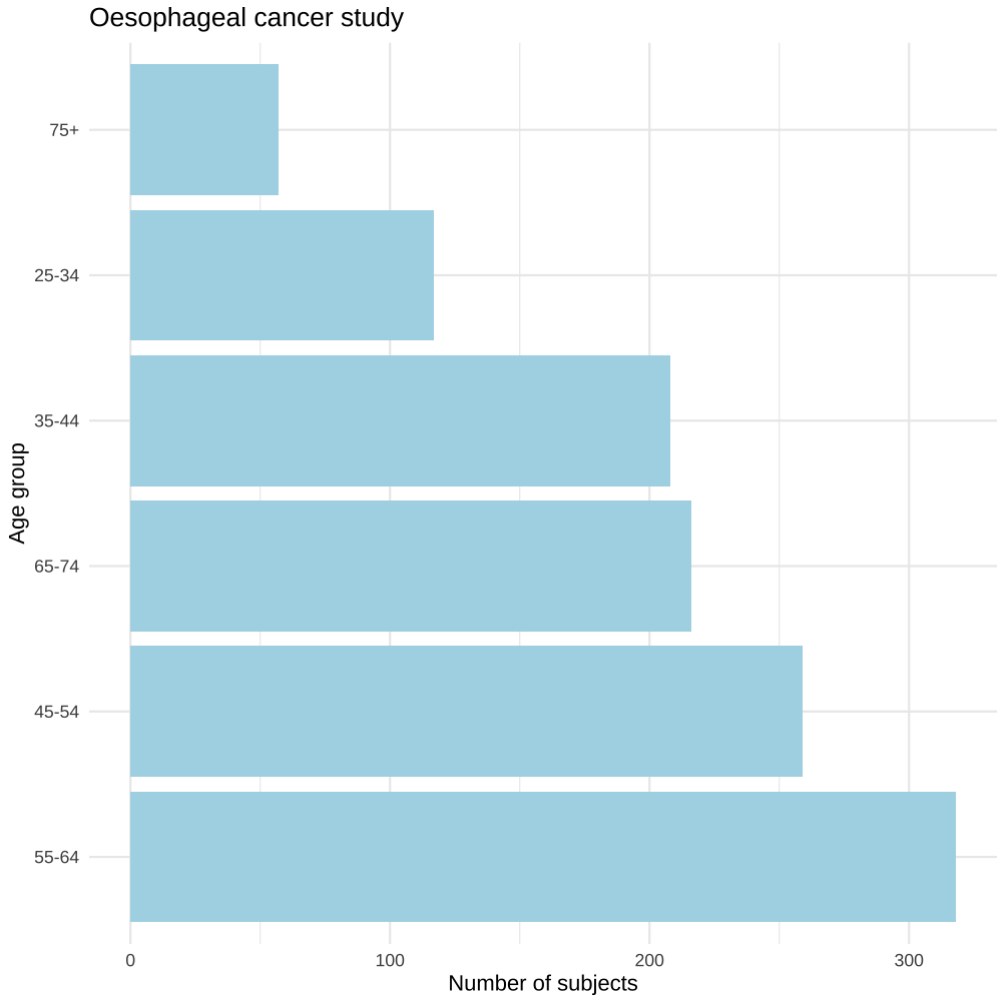
Substitute `agegp` for the x-axis


Oesophageal cancer study
75+
65-74
55-64
45-54
35-44
25-34
Age group
0
100
200
300
Number of subjects

Re-order the x-axis by descending number of subjects

- bad!
  - `agegrp` has a natural order, we shouldn't mess with it
  - but just as an exercise...
- note that `agegp` is a factor variable
  - hint: see `fct_reorder()` in the `forcats` package (part of `tidyverse` meta-package)

```
esoph_ca %>%
  group_by(agegp) %>%
  summarise(nsubjects = sum(ncases + ncontrols)) %>%
  ggplot(aes(x = fct_reorder(agegp, desc(nsubjects)),
             y = nsubjects)) +
    geom_col(fill = "lightblue") +
    labs(x="Age group",
         y="Number of subjects",
         title="Oesophageal cancer study") +
    coord_flip() +
    theme_minimal()
```


Oesophageal cancer study
75+
25-34
35-44
65-74
45-54
55-64
Age group
0
100
200
300
Number of subjects

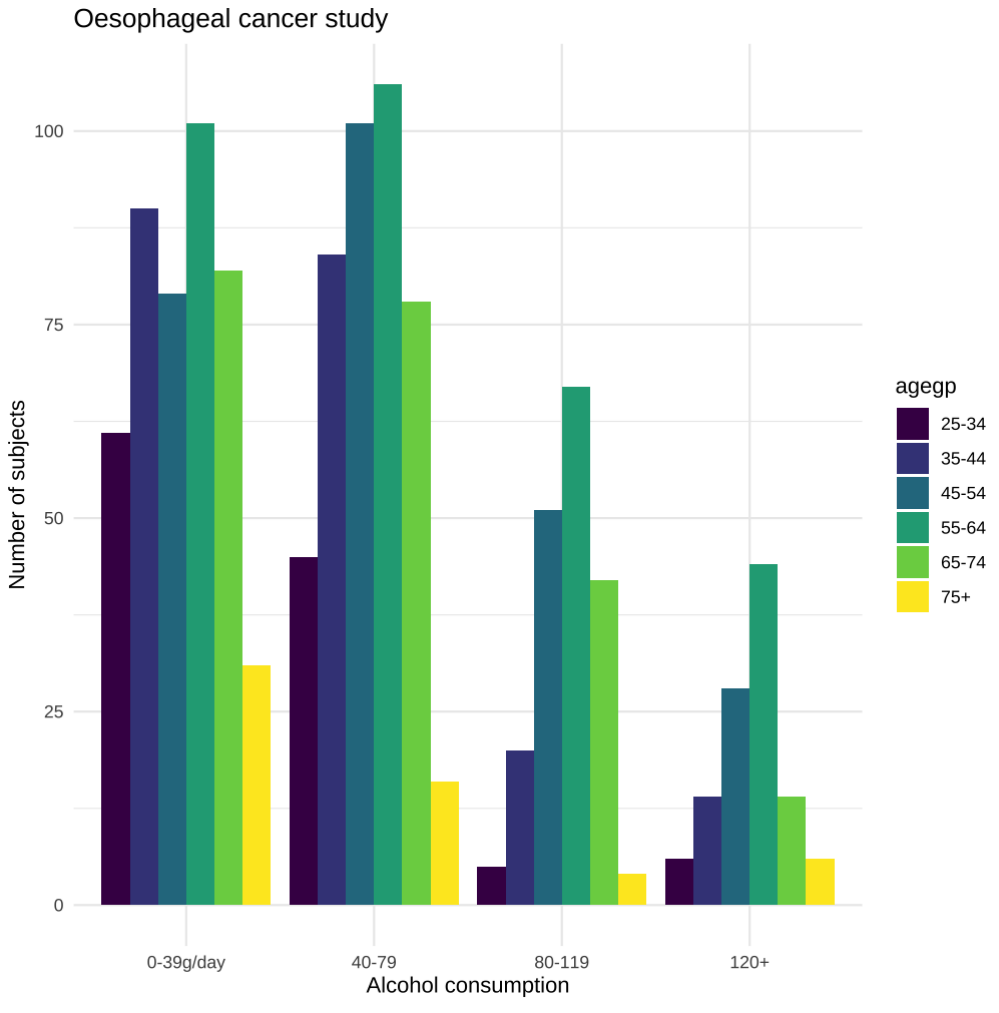
Section 6.2 - grouped and stacked bars

Grouped bars

- write some R code using `ggplot2` to make a bar chart using the `esoph_ca` dataset in the `medicaldata` package to create a bar chart which shows:
  - `nsubjects` on the y-axis (you'll need to add `ncases` and `ncontrols` to create an `nsubjects` column using `dplyr`
  - `alcgrp` on the x-axis
  - grouped bars for each value of `agegp`
- hint: the `position` argument to `geom_col()` or `geom_bar()`

```
esoph_ca %>%
  group_by(alcgp, agegp) %>%
  summarise(nsubjects = sum(ncases + ncontrols)) %>%
  ggplot(aes(x = alcgp, y = nsubjects, fill = agegp)) +
    geom_col(position = "dodge") +
    labs(x="Alcohol consumption",
         y="Number of subjects",
         title="Oesophageal cancer study") +
    theme_minimal()
```

```
## `summarise()` has grouped output by 'alcgp'. You can override using the
## `.groups` argument.
```

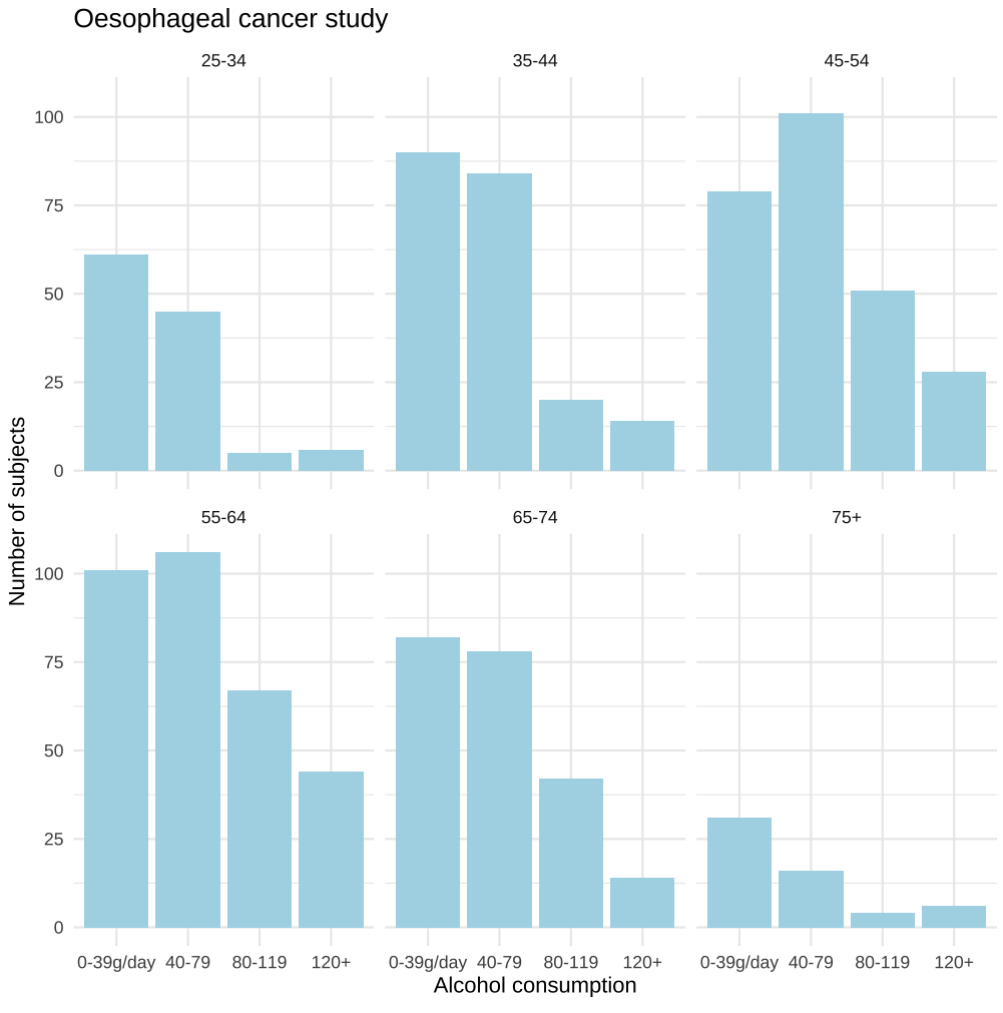


Change to facetting by `agegrp` with just one fill colour

- as suggested by Wilke

```
esoph_ca %>%
  group_by(alcgp, agegp) %>%
  summarise(nsubjects = sum(ncases + ncontrols)) %>%
  ggplot(aes(x = alcgp, y = nsubjects)) +
    geom_col(fill="lightblue") +
    facet_wrap(~agegp) +
    labs(x="Alcohol consumption",
         y="Number of subjects",
         title="Oesophageal cancer study") +
    theme_minimal()
```

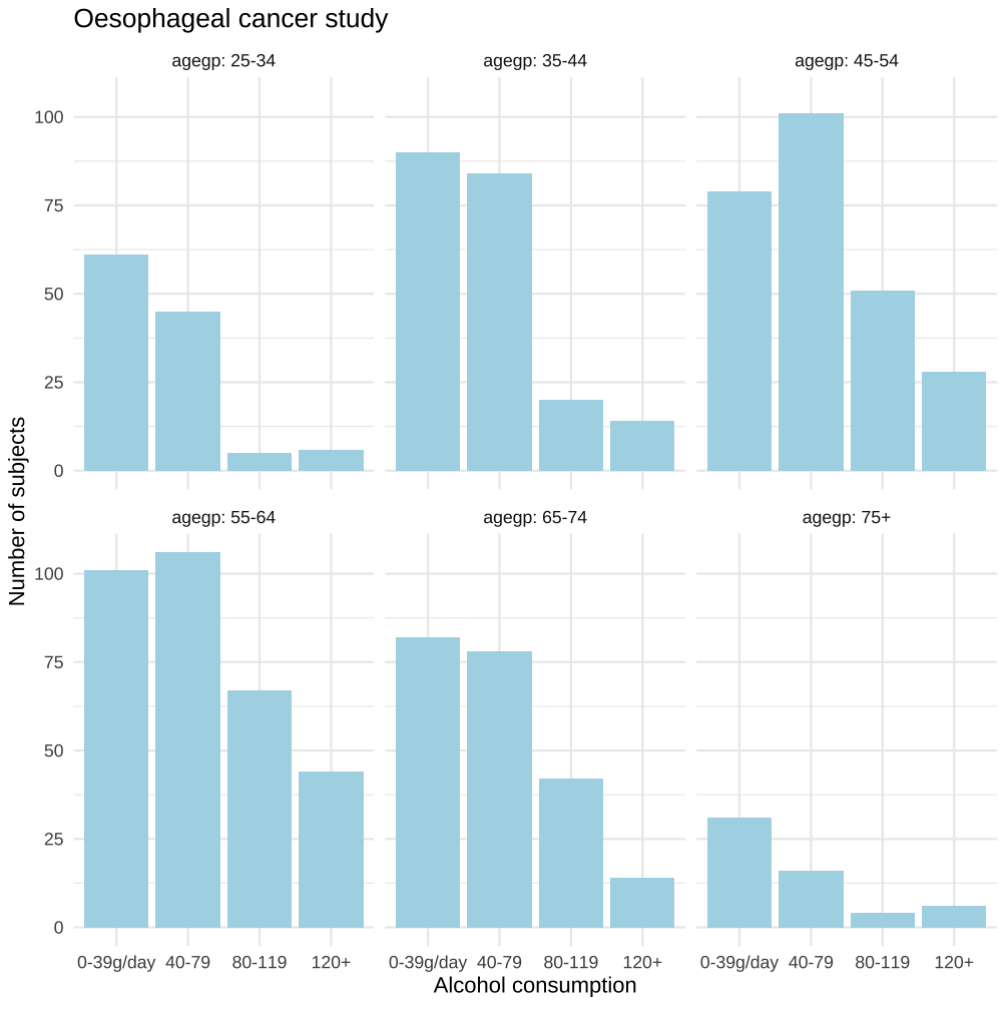
```
## `summarise()` has grouped output by 'alcgp'. You can override using the
## `.groups` argument.
```



Can you label the facets to show they are age group?

- hint: the `labeller=` argument to `facet_wrap()` and `facet_grid()`

```
## `summarise()` has grouped output by 'alcgp'. You can override using the
## `.groups` argument.
```



Let's stack the bars by whether they are a case or control

- we need to reshape our dataset
  - one column holding the count of subjects
  - one column holding whether the count is for cases or control
- hint: the `pivot_longer()` function the `tidyr` package (part of the `tidyverse`)

Data tidying with tidyr : : CHEAT SHEET

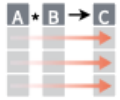
Tidy data is a way to organize tabular data in a consistent data structure across packages.
A table is tidy if:

Each **variable** is in its own **column**

&

Each **observation**, or **case**, is in its own row

Access **variables** as **vectors**



Preserve **cases** in vectorized operations

Tibbles

AN ENHANCED DATA FRAME

Tibbles are a table format provided by the **tibble** package. They inherit the data frame class, but have improved behaviors:

- **Subset** a new tibble with], a vector with [[and $.
- **No partial matching** when subsetting columns.
- **Display** concise views of the data on one screen.

options(tibble.print_max = n, tibble.print_min = m, tibble.width = Inf**)** Control default display settings.

View() or **glimpse()** View the entire data set.

CONSTRUCT A TIBBLE

tibble(…) Construct by columns.
tibble(x = 1:3, y = c("a", "b", "c"))

tribble(…) Construct by rows.
tribble(~x, ~y,
1, "a",
2, "b",

Both make this tibble

```
A tibble: 3 × 2
      x     y
  <int> <chr>
1     1     a
2     2     b
```

Reshape Data - Pivot data to reorganize values into a new layout.

table4a

country	1999	2000
A	0.7K	2K
B	37K	80K
C	212K	213K

→

country	year	cases
A	1999	0.7K
B	1999	37K
C	1999	212K
A	2000	2K
B	2000	80K
C	2000	213K

pivot_longer(data, cols, names_to = "name", values_to = "value", values_drop_na = FALSE**)**

"Lengthen" data by collapsing several columns into two. Column names move to a new names_to column and values to a new values_to column.

pivot_longer(table4a, cols = 2:3, names_to ="year", values_to = "cases")

table2

country	year	type	count
A	1999	cases	0.7K
A	1999	pop	19M
A	2000	cases	2K
A	2000	pop	20M
B	1999	cases	37K
B	1999	pop	172M
B	2000	cases	80K
B	2000	pop	174M
C	1999	cases	212K
C	1999	pop	1T
C	2000	cases	213K
C	2000	pop	1T

→

country	year	cases	pop
A	1999	0.7K	19M
A	2000	2K	20M
B	1999	37K	172M
B	2000	80K	174M
C	1999	212K	1T
C	2000	213K	1T

pivot_wider(data, names_from = "name", values_from = "value"**)**

The inverse of pivot_longer(). "Widen" data by expanding two columns into several. One column provides the new column names, the other the values.

pivot_wider(table2, names_from = type, values_from = count)

Split Cells - Use these functions to split or combine cells into individual, isolated values.

table5

country	century	year
A	19	99
A	20	00
B	19	99
B	20	00

→

country	year
A	1999
A	2000
B	1999
B	2000

unite(data, col, …, sep = "_", remove = TRUE, na.rm = FALSE**)** Collapse cells across several columns into a single column.

unite(table5, century, year, col = "year", sep = "")

table3

country	year	rate
A	1999	0.7K/19M
A	2000	2K/20M
B	1999	37K/172M
B	2000	80K/174M

→

country	year	cases	pop
A	1999	0.7K	19M
A	2000	2K	20M
B	1999	37K	172
B	2000	80K	174

separate(data, col, into, sep = "[^[:alnum:]]+", remove = TRUE, convert = FALSE, extra = "warn", fill = "warn", …**)** Separate each cell in a column into several columns. Also **extract()**.

separate(table3, rate, sep = "/"

Expand Tables

Create new combinations of variables or identify implicit missing values (combinations of variables not present in the data).

x

x1	x2	x3
A	1	3
B	1	4
B	2	3

→

x1	x2
A	1
A	2
B	1
B	2

expand(data, …**)** Create a new tibble with all possible combinations of the values of the variables listed in … Drop other variables.
expand(mtcars, cyl, gear, carb)

x

x1	x2	x3
A	1	3
B	1	4
B	2	3

→

x1	x2	x3
A	1	3
A	2	NA
B	1	4
B	2	3

complete(data, …, fill = list()**)** Add missing possible combinations of values of variables listed in … Fill remaining variables with NA.
complete(mtcars, cyl, gear, carb)

Handle Missing Values

Drop or replace explicit missing values (NA).

x

x1	x2
A	1
B	NA
C	NA
D	3
E	NA

→

x1	x2
A	1
D	3

drop_na(data, …**)** Drop rows containing NA's in … columns.
drop_na(x, x2)

x

x1	x2
A	1
B	NA
C	NA
D	3
E	NA

→

x1	x2
A	1
B	1
C	1
D	3
E	3

fill(data, …, .direction = "down"**)** Fill in NA's in … columns using the next or previous value.

Reshape Data - Pivot data to reorganize values into a new layout.

table4a

country	1999	2000
A	0.7K	2K
B	37K	80K
C	212K	213K

country	year	cases
A	1999	0.7K
B	1999	37K
C	1999	212K
A	2000	2K
B	2000	80K
C	2000	213K

pivot_longer(data, cols, names_to = "name", values_to = "value", values_drop_na = FALSE**)**

"Lengthen" data by collapsing several columns into two. Column names move to a new names_to column and values to a new values_to column.

```
pivot_longer(table4a, cols = 2:3, names_to ="year",
  values_to = "cases")
```

table2

Step by step!

```
esoph_ca
```

```
##    agegp     alcgp    tobgp ncases ncontrols
## 1  25-34 0-39g/day 0-9g/day      0        40
## 2  25-34 0-39g/day    10-19      0        10
## 3  25-34 0-39g/day    20-29      0         6
## 4  25-34 0-39g/day      30+      0         5
## 5  25-34     40-79 0-9g/day      0        27
## 6  25-34     40-79    10-19      0         7
## 7  25-34     40-79    20-29      0         4
## 8  25-34     40-79      30+      0         7
## 9  25-34    80-119 0-9g/day      0         2
## 10 25-34    80-119    10-19      0         1
## 11 25-34    80-119      30+      0         2
## 12 25-34      120+ 0-9g/day      0         1
## 13 25-34      120+    10-19      1         1
## 14 25-34      120+    20-29      0         1
## 15 25-34      120+      30+      0         2
## 16 35-44 0-39g/day 0-9g/day      0        60
## 17 35-44 0-39g/day    10-19      1        14
```

```
esoph_ca %>%
  group_by(alcgp, agegp) %>%
  summarise(ncases = sum(ncases),
            ncontrols = sum(ncontrols))
```

```
## `summarise()` has grouped output by 'alcgp'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 24 × 4
## # Groups:   alcgp [4]
##    alcgp     agegp ncases ncontrols
##    <ord>     <ord>  <dbl>     <dbl>
##  1 0-39g/day 25-34      0        61
##  2 0-39g/day 35-44      1        89
##  3 0-39g/day 45-54      1        78
##  4 0-39g/day 55-64     12        89
##  5 0-39g/day 65-74     11        71
##  6 0-39g/day 75+        4        27
##  7 40-79     25-34      0        45
##  8 40-79     35-44      4        80
##  9 40-79     45-54     20        81
## 10 40-79     55-64     22        84
## # … with 14 more rows
```

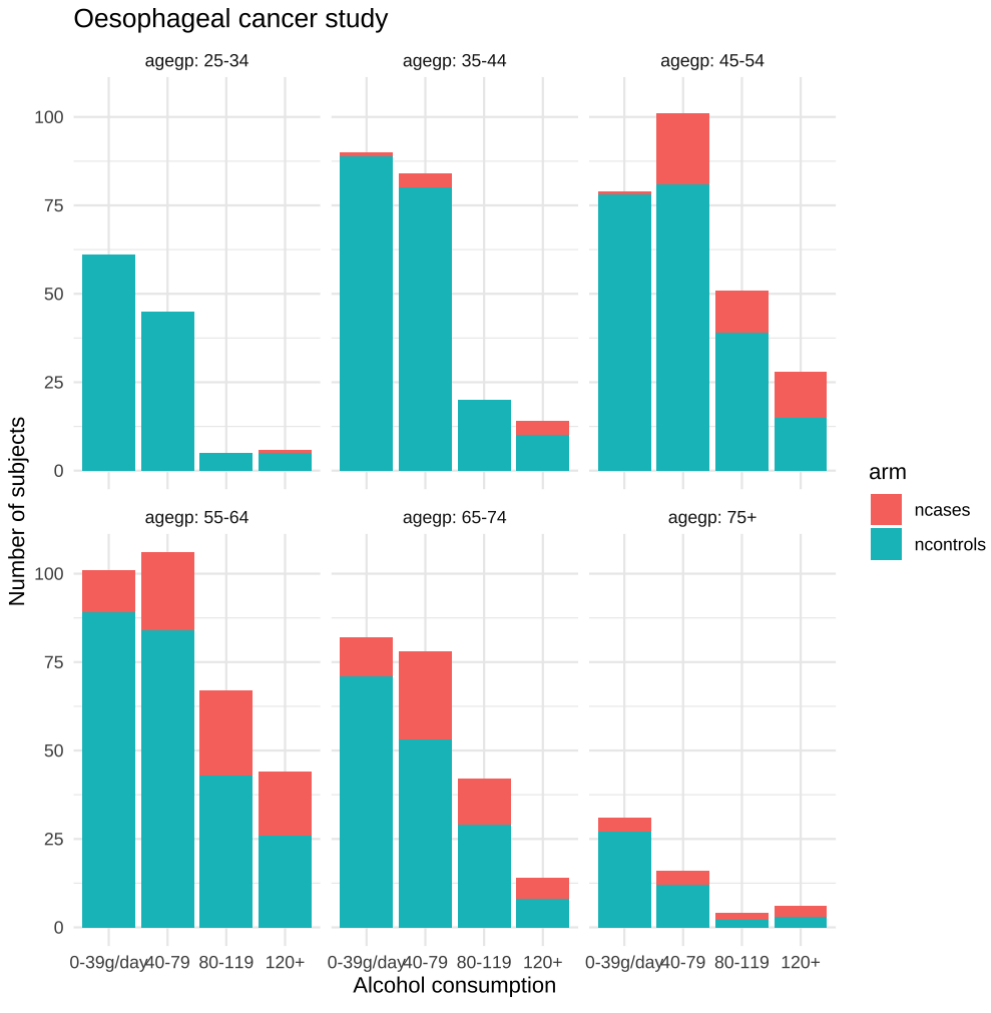
```
esoph_ca %>%
  group_by(alcgp, agegp) %>%
  summarise(ncases = sum(ncases),
            ncontrols = sum(ncontrols)) %>%
  pivot_longer(cols = c(ncases, ncontrols),
               names_to = "arm",
               values_to = "nsubjects")
```

```
## `summarise()` has grouped output by 'alcgp'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 48 × 4
## # Groups:   alcgp [4]
##    alcgp     agegp arm       nsubjects
##    <ord>     <ord> <chr>         <dbl>
##  1 0-39g/day 25-34 ncases            0
##  2 0-39g/day 25-34 ncontrols        61
##  3 0-39g/day 35-44 ncases            1
##  4 0-39g/day 35-44 ncontrols        89
##  5 0-39g/day 45-54 ncases            1
##  6 0-39g/day 45-54 ncontrols        78
##  7 0-39g/day 55-64 ncases           12
##  8 0-39g/day 55-64 ncontrols        89
##  9 0-39g/day 65-74 ncases           11
## 10 0-39g/day 65-74 ncontrols        71
```

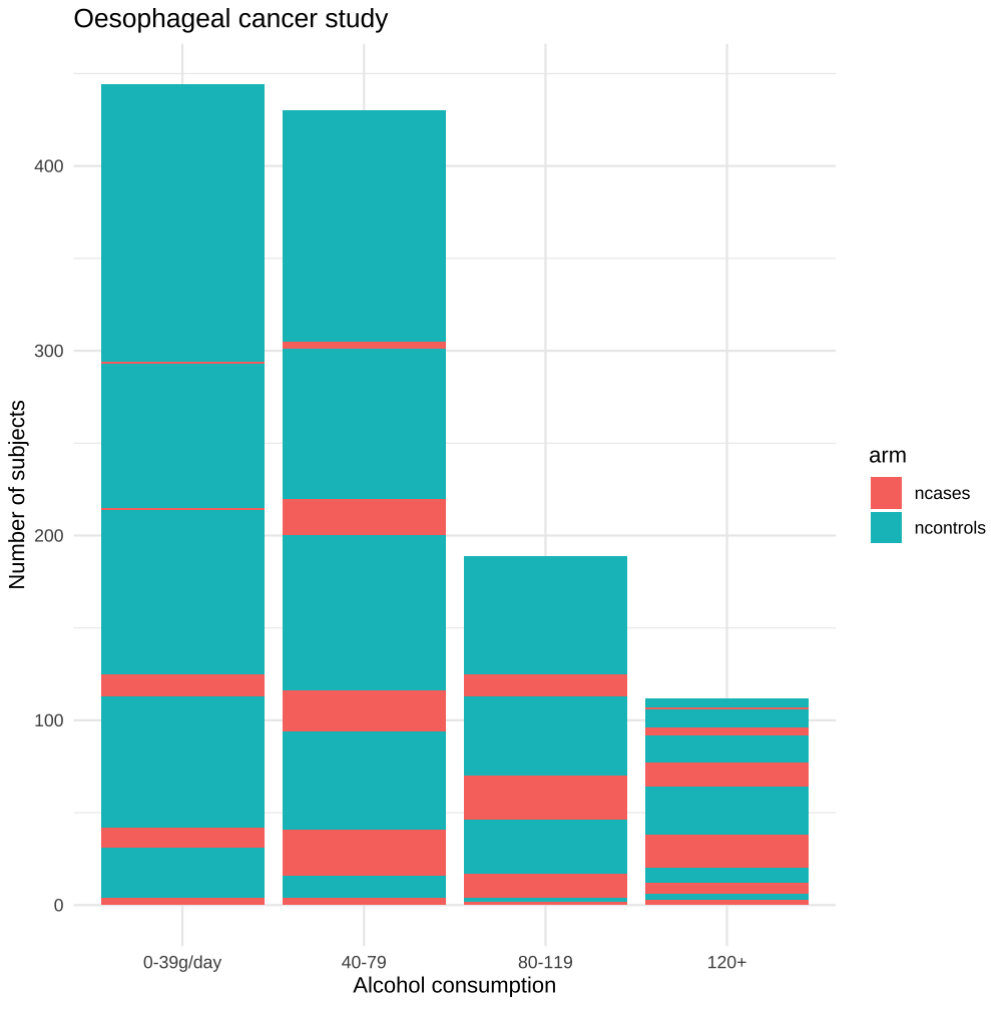
```r
esoph_ca %>%
  group_by(alcgp, agegp) %>%
  summarise(ncases = sum(ncases),
            ncontrols = sum(ncontrols)) %>%
  pivot_longer(cols = c(ncases, ncontrols),
               names_to = "arm",
               values_to = "nsubjects") %>%
  ggplot(aes(x = alcgp, y = nsubjects, fill = arm)) +
    geom_col() +
    facet_wrap(~agegp, labeller = label_both) +
    labs(x="Alcohol consumption",
         y="Number of subjects",
         title="Oesophageal cancer study") +
    theme_minimal()
```

```
## `summarise()` has grouped output by 'alcgp'. You can override using the
## `.groups` argument.
```



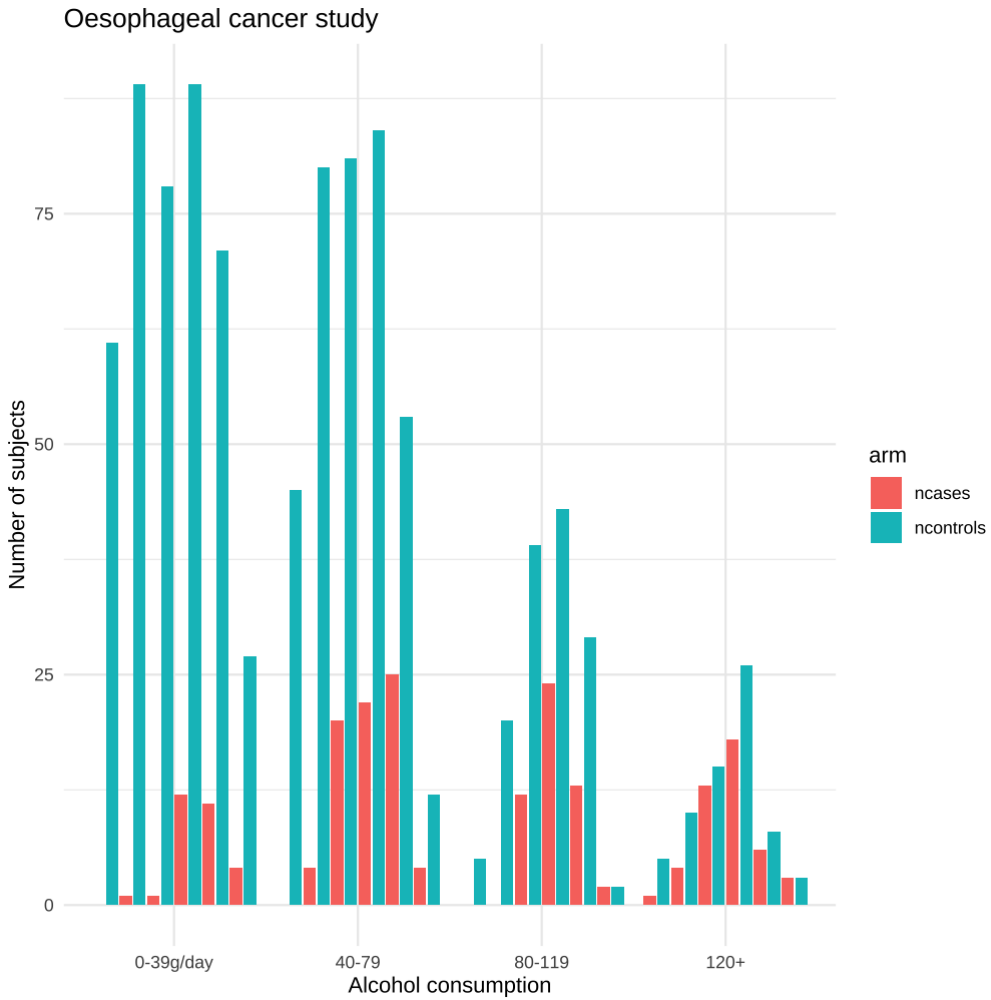
```
esoph_ca %>%
  group_by(alcgp, agegp) %>%
  summarise(ncases = sum(ncases),
            ncontrols = sum(ncontrols)) %>%
  pivot_longer(cols = c(ncases, ncontrols),
               names_to = "arm",
               values_to = "nsubjects") %>%
  ggplot(aes(x = alcgp, y = nsubjects, fill = arm, group = agegp)) +
    geom_col() +
    labs(x="Alcohol consumption",
         y="Number of subjects",
         title="Oesophageal cancer study") +
    theme_minimal()
```

```
## `summarise()` has grouped output by 'alcgp'. You can override using the
## `.groups` argument.
```



```
esoph_ca %>%
  group_by(alcgp, agegp) %>%
  summarise(ncases = sum(ncases),
            ncontrols = sum(ncontrols)) %>%
  pivot_longer(cols = c(ncases, ncontrols),
               names_to = "arm",
               values_to = "nsubjects") %>%
  ggplot(aes(x = alcgp, y = nsubjects, fill = arm, group = agegp)) +
    geom_col(position = "dodge2") +
    labs(x="Alcohol consumption",
         y="Number of subjects",
         title="Oesophageal cancer study") +
    theme_minimal()
```

```
## `summarise()` has grouped output by 'alcgp'. You can override using the
## `.groups` argument.
```

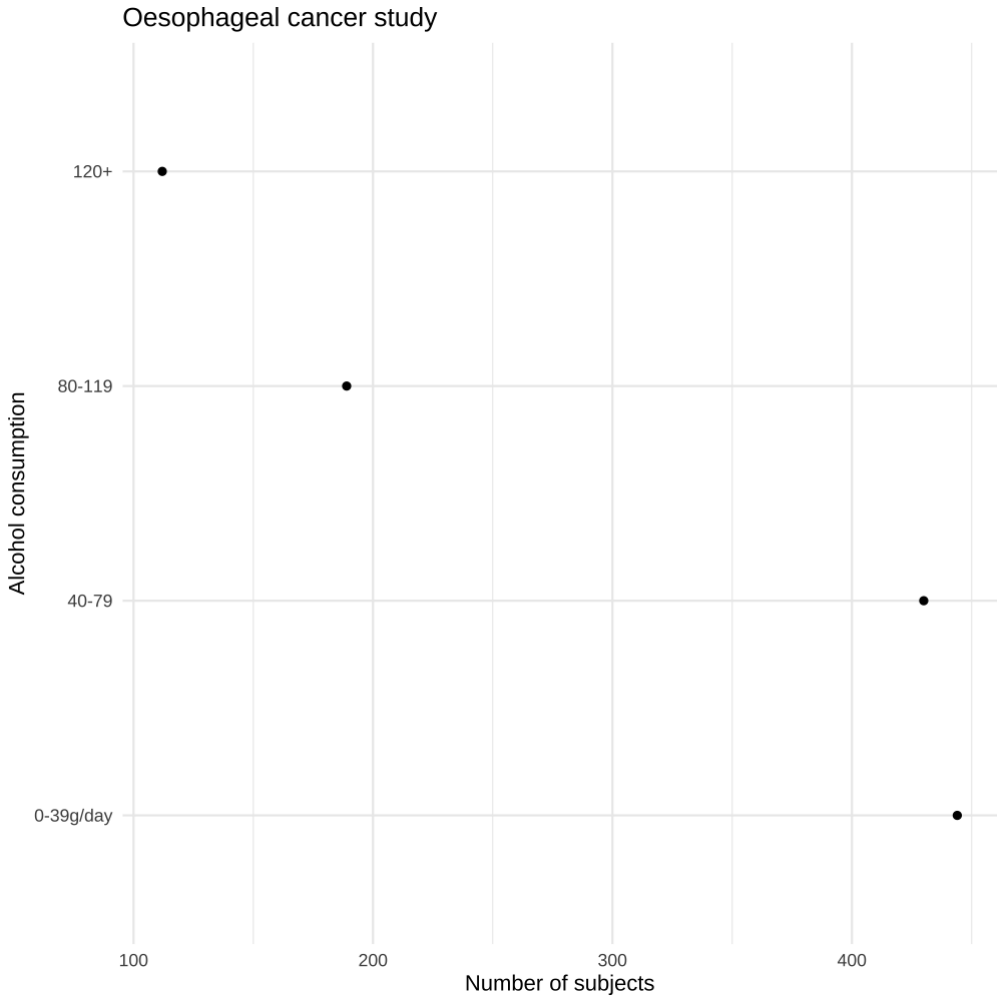


Section 6.3 Dotplots and heatmaps

Draw a dotplot

- using the `esoph_ca` dataset in the `medicaldata` package to create a dot plot which shows:
  - `nsubjects` on the y-axis (you'll need to add `ncases` and `ncontrols` to create an `nsubjects` column using `dplyr`
  - `alcgrp` on the x-axis
  - fill the co-ordinates by 90 degrees
- hint: not `geom_dotplot()`!

```
esoph_ca %>%
  group_by(alcgp) %>%
  summarise(nsubjects = sum(ncases + ncontrols)) %>%
  ggplot(aes(x = alcgp, y = nsubjects)) +
    geom_point(fill="lightblue") +
    labs(x="Alcohol consumption",
         y="Number of subjects",
         title="Oesophageal cancer study") +
    theme_minimal() +
    coord_flip()
```

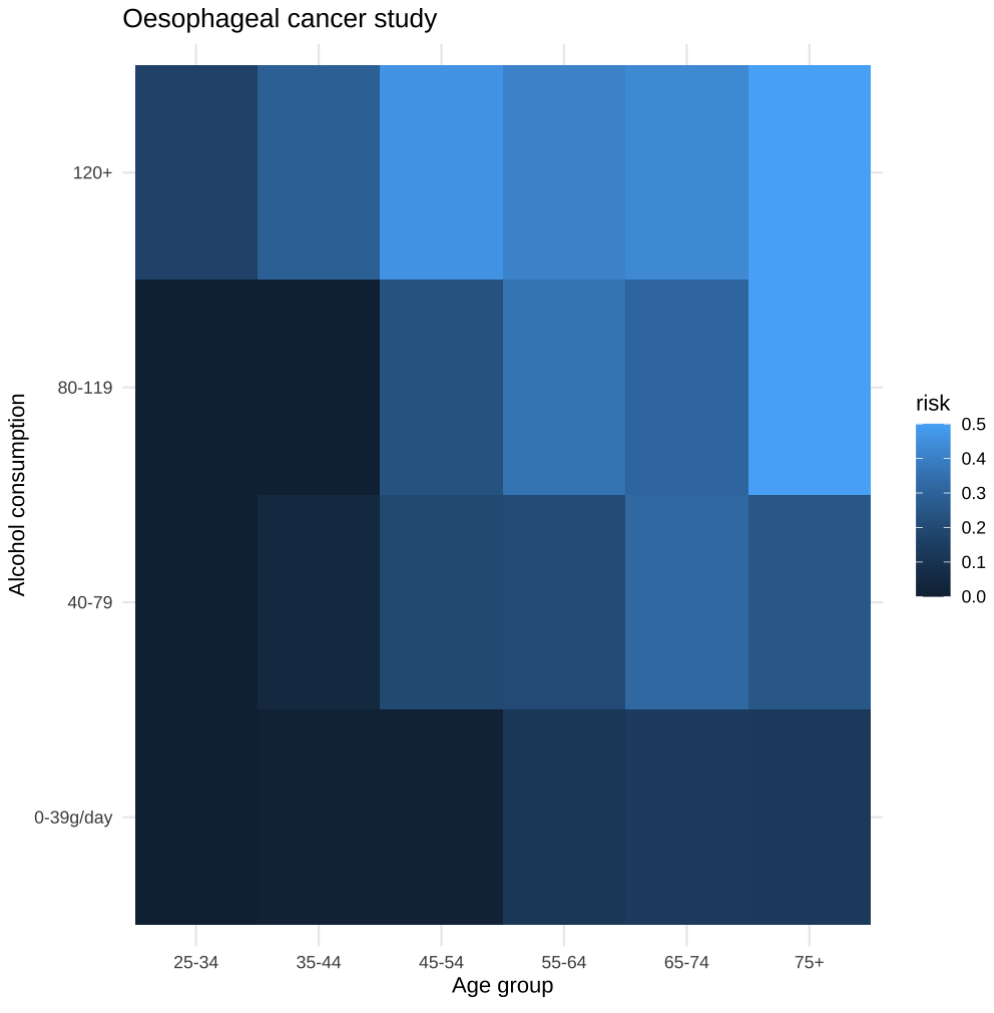

Oesophageal cancer study
120+
80-119
40-79
0-39g/day
Alcohol consumption
100
200
300
400
Number of subjects

Heatmap of oesophageal cancer risk

- summarise the `esoph_ca` so that we have the proportion of cases (that is, `ncases/(ncases + ncontrols)`) as a heatmap by `agegp` and `alcgp`
- hint: `geom_tile()`

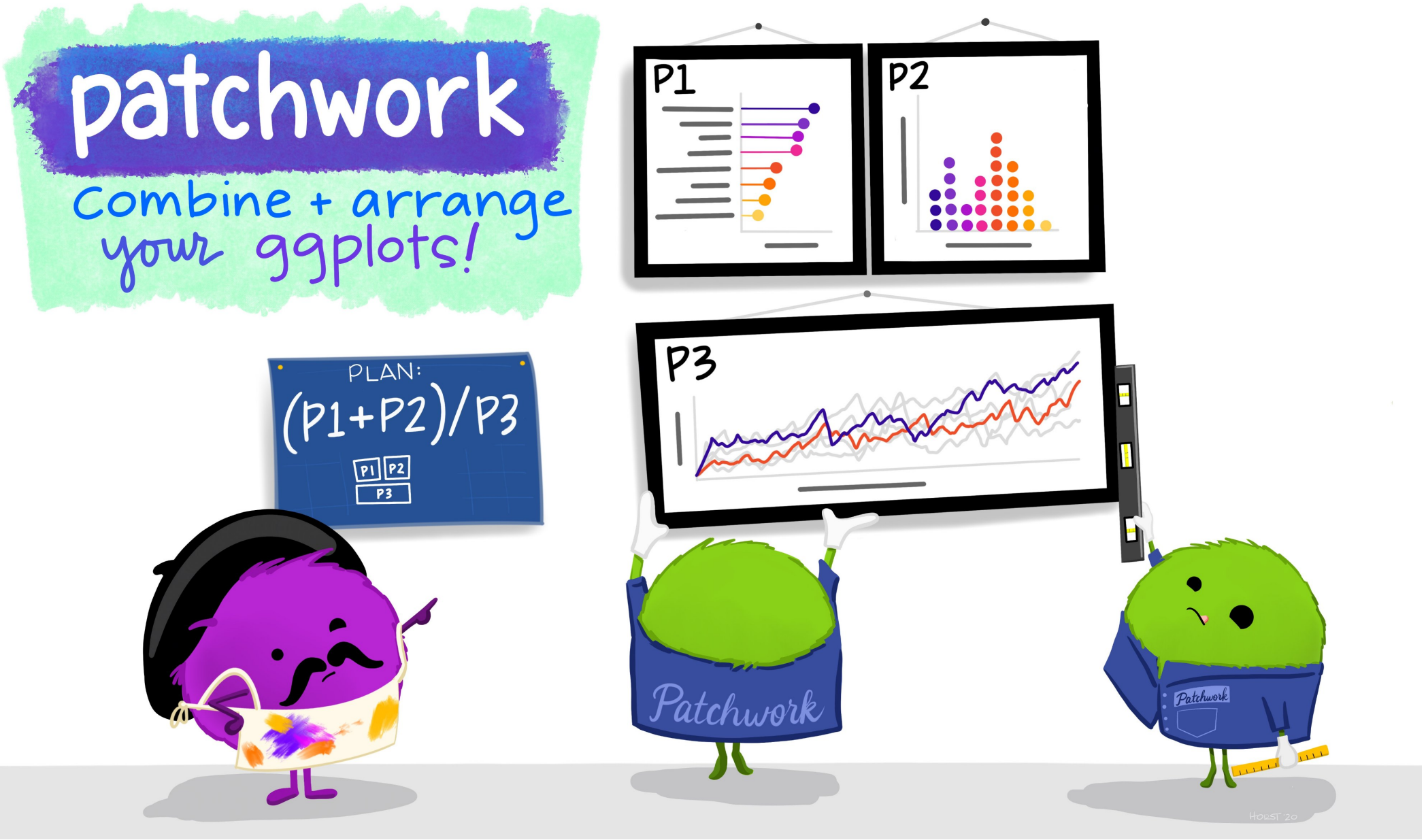
```
esoph_ca %>%
  group_by(alcgp, agegp) %>%
  summarise(risk = sum(ncases) / sum(ncases + ncontrols)) %>%
  ggplot(aes(x = agegp, y = alcgp, fill = risk)) +
    geom_tile() +
    labs(x="Age group",
         y="Alcohol consumption",
         title="Oesophageal cancer study") +
    theme_minimal()
```

```
## `summarise()` has grouped output by 'alcgp'. You can override using the
## `.groups` argument.
```



Challenge

- improve the colour palette and legend


patchwork
Combine + arrange your ggplots!
PLAN:
(P1+P2)/P3
P1
P2
P3
P1
P2
P3
Patchwork
Patchwork
HORST '20

https://patchwork.data-imaginist.com