Creating plots in R using ggplot2 - part 3: bar plots

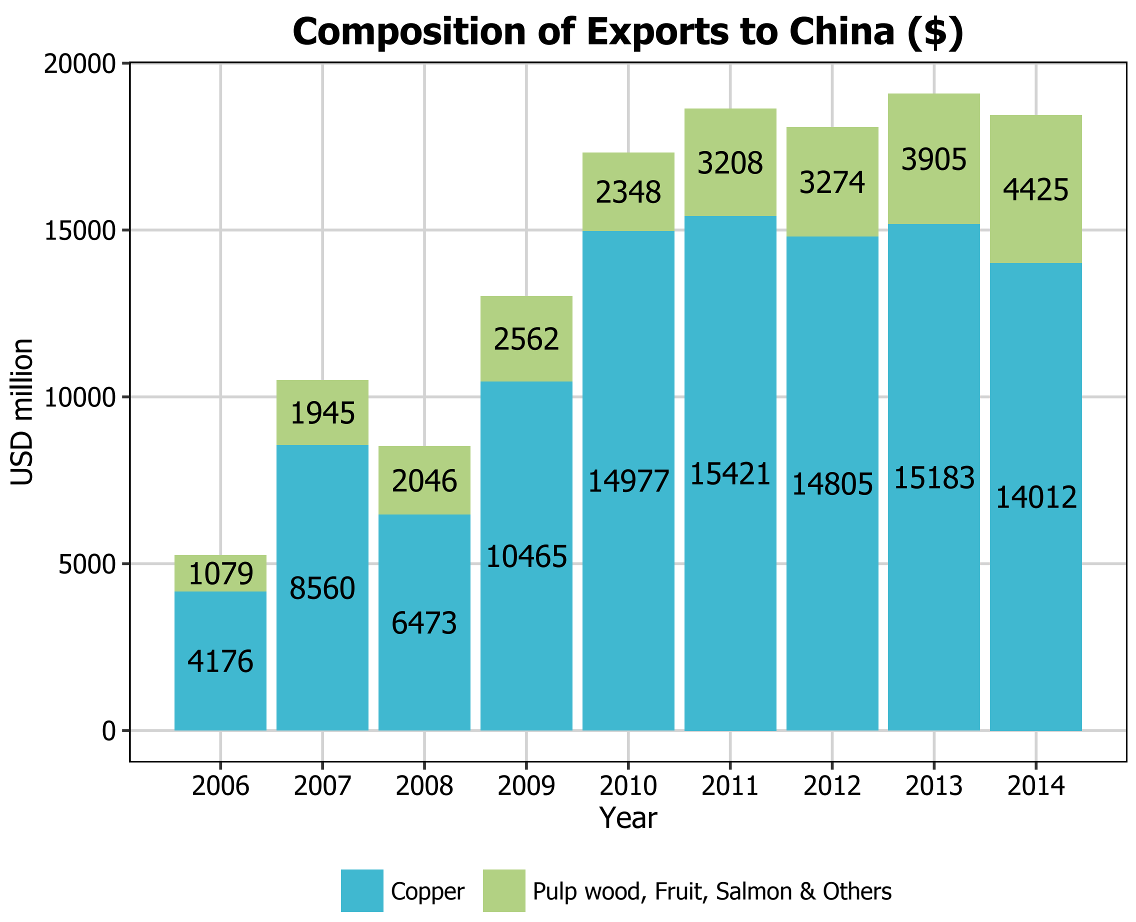
Jodie Burchell
Mauricio Vargas Sepúlveda

2016-05-13

Contents

In this third tutorial I am doing with Mauricio Vargas Sepúlveda, we will demonstrate some of the many options the ggplot2 package has for creating and customising bar plots. We will use the same dataset from the first post.

In this tutorial, we will work towards creating the area plot below. We will take you from a basic bar plot and explain all the customisations we add to the code step-by-step.



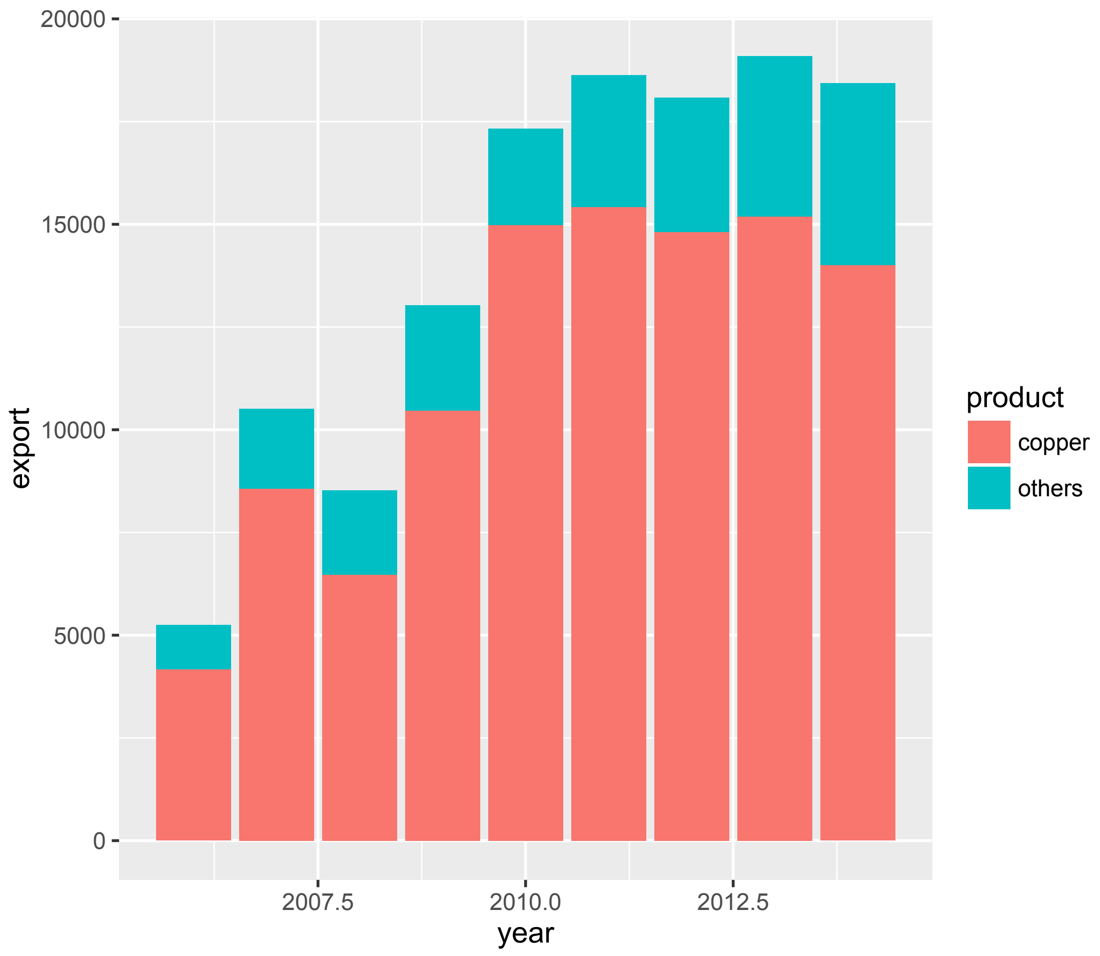
Basic graph

The first thing to do is load in the data and the libraries, as below:

```
library(ggplot2)
library(ggthemes)
library(extrafont)
library(plyr)
library(scales)

charts.data <- read.csv("copper-data-for-tutorial.csv")
```

In order to initialise a plot we tell ggplot that `charts.data` is our data, and specify the variables on each axis. We then instruct ggplot to render this as an bar plot by adding the `geom_area` command.

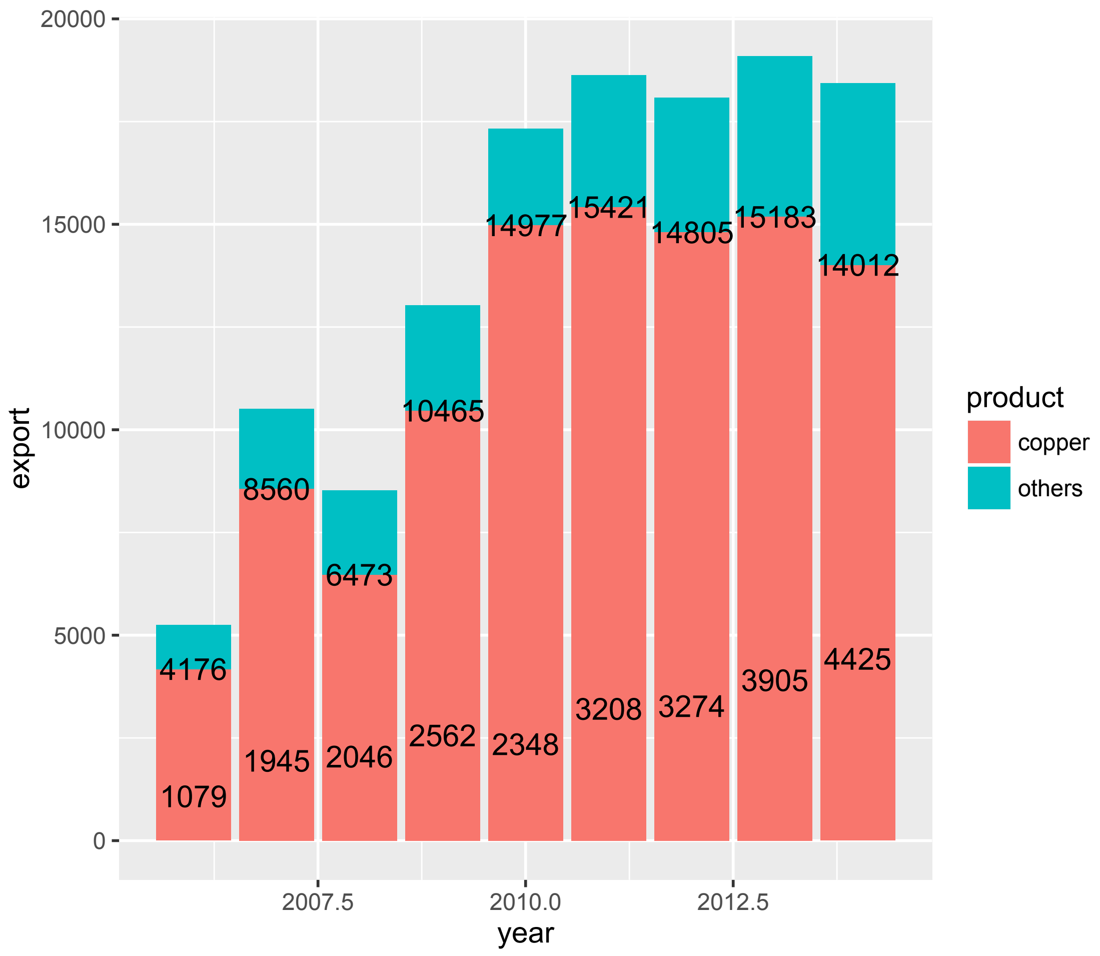
```
p3 <- ggplot() + geom_bar(aes(y = export, x = year, fill = product), data = charts.data, stat="identity
p3
```



Adding data labels

To label the bars according to some variable in the data, we add the `label` argument to the `ggplot(aes())` option. In this case, we have labelled the bars with numbers from the `export` variable.

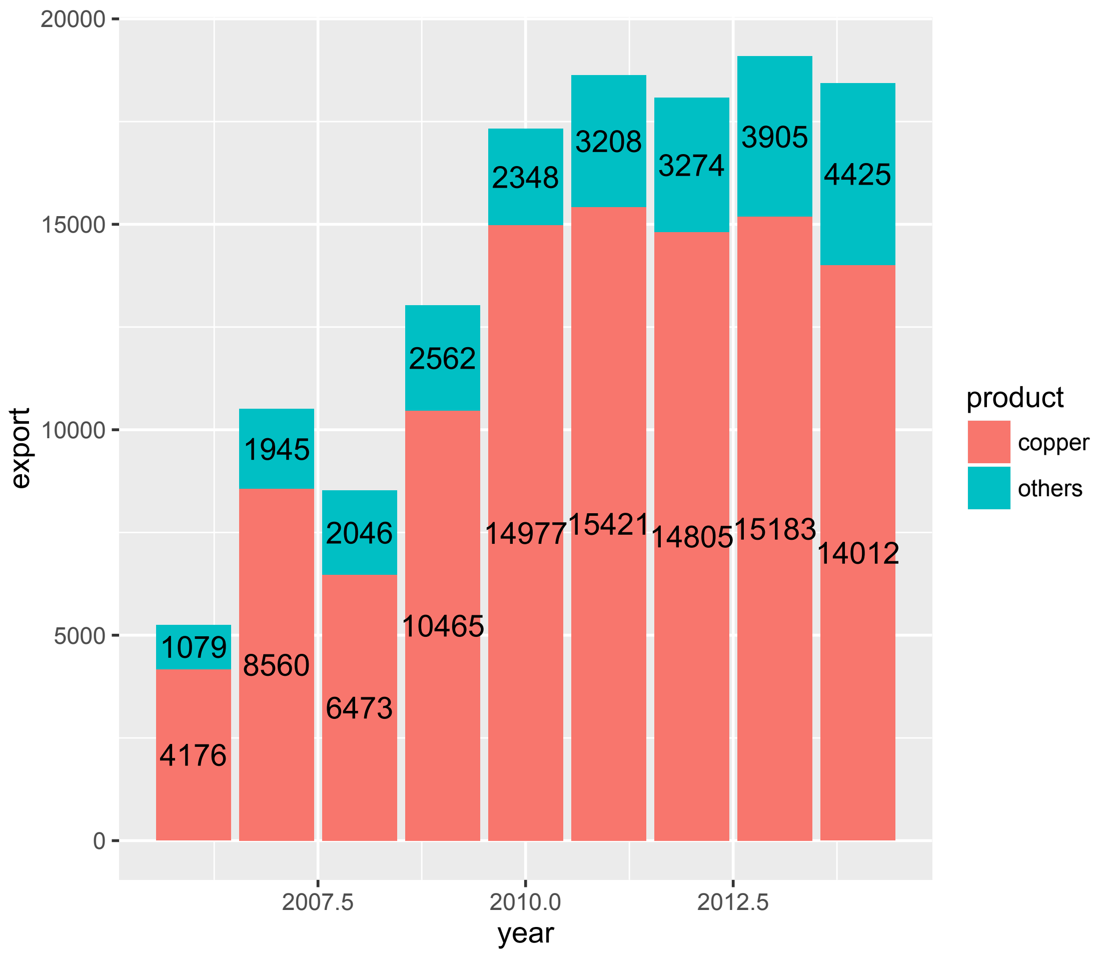
```
p3 <- p3 + geom_text(data=charts.data, aes(x = year, y = export, label = export), size=4)
p3
```



Adjusting data labels position

To adjust the position of the data labels from the default placement, we use the `ddply` function on the data, and create a new variable called `pos`. This variable is at the centre of each bar and can be used to specify the position of the labels by assigning it to the `y` argument in `geom_text(aes())`.

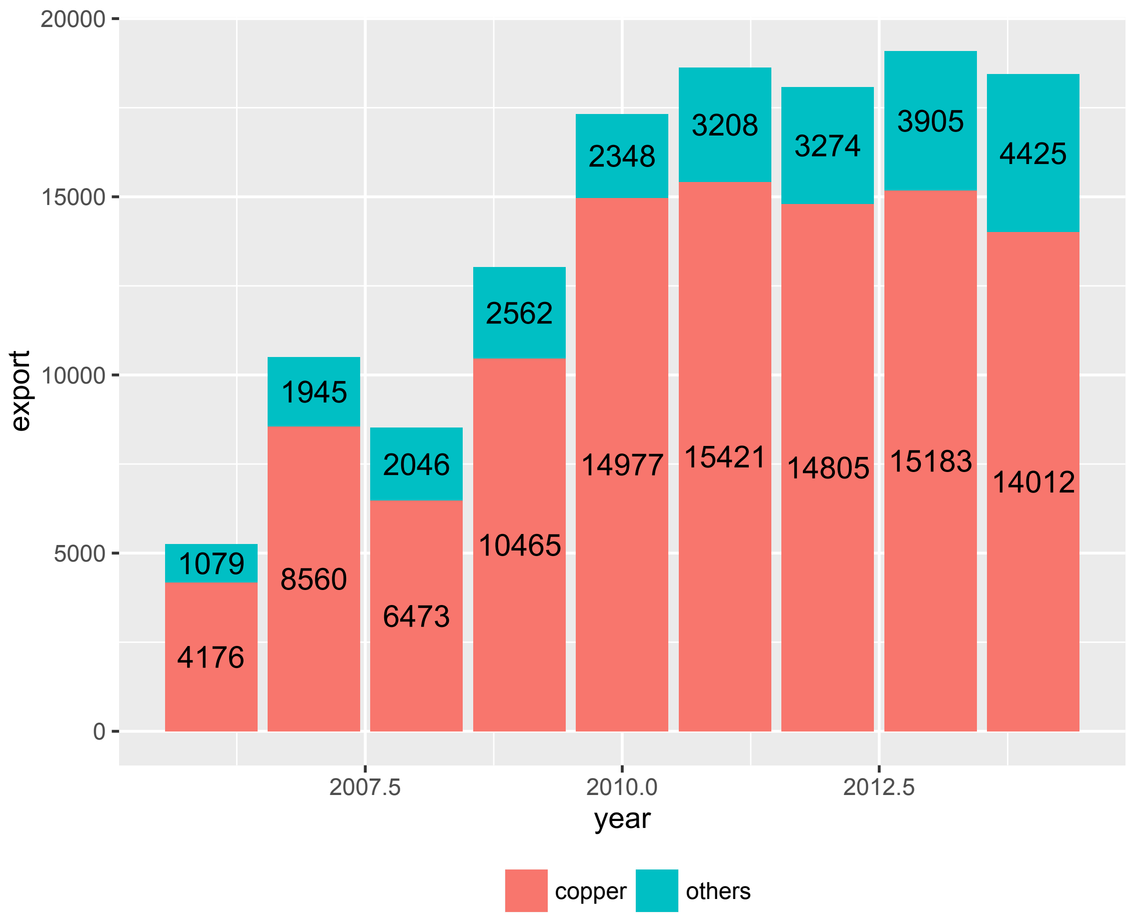
```
charts.data <- ddply(charts.data, .(year), transform, pos = cumsum(export) - (0.5 * export))

p3 <- ggplot() + geom_bar(aes(y = export, x = year, fill = product), data = charts.data, stat="identity
p3 <- p3 + geom_text(data=charts.data, aes(x = year, y = pos, label = export), size=4)
p3
```



Adjusting legend position

To adjust the position of the legend from the default spot of right of the graph, we add the `theme` option and specify the `legend.position="bottom"` argument. We can also change the title to blank using the `legend.title = element_blank()` argument and change the legend shape using the `legend.direction="horizontal"` argument.

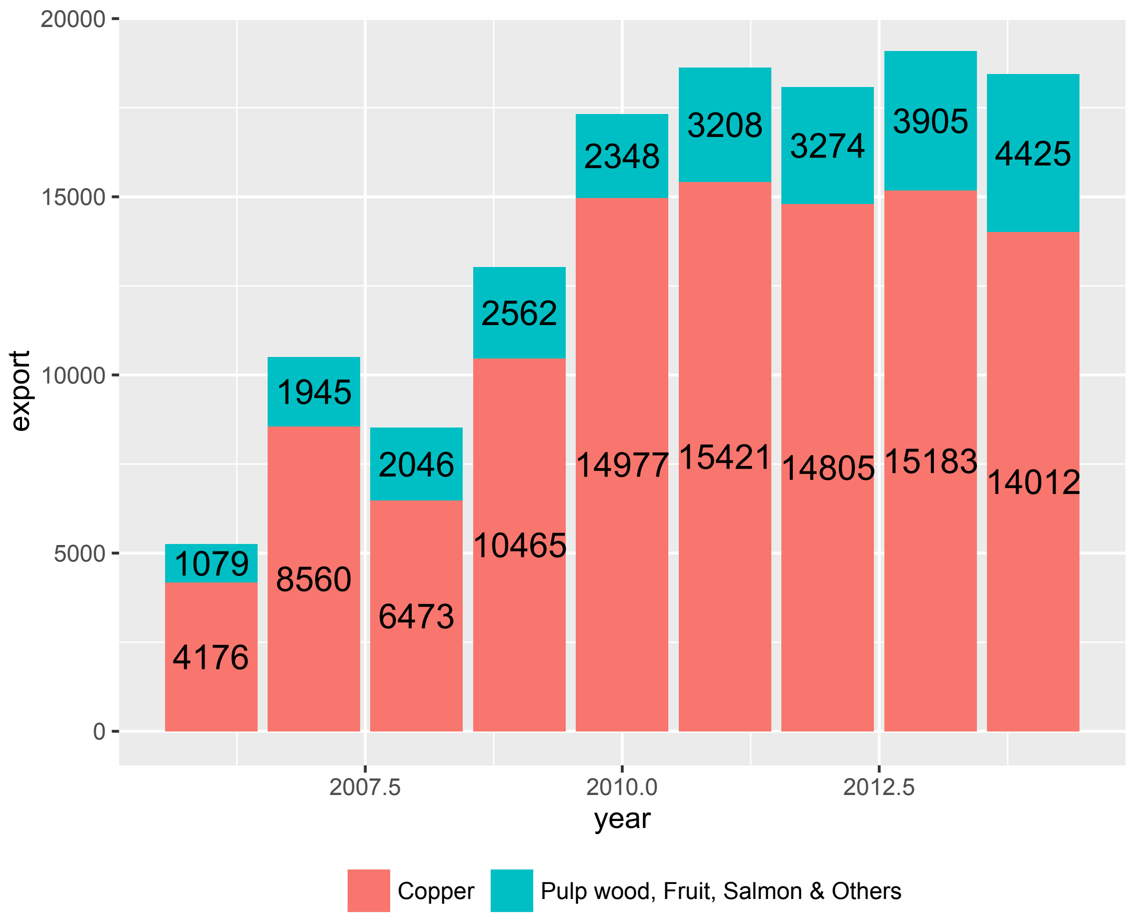
```
p3 <- p3 + theme(legend.position="bottom", legend.direction="horizontal",
  legend.title = element_blank())
p3
```



Changing variables display

To change the variables' displayed name, we need to re-factor our data labels in `charts.data` data frame.

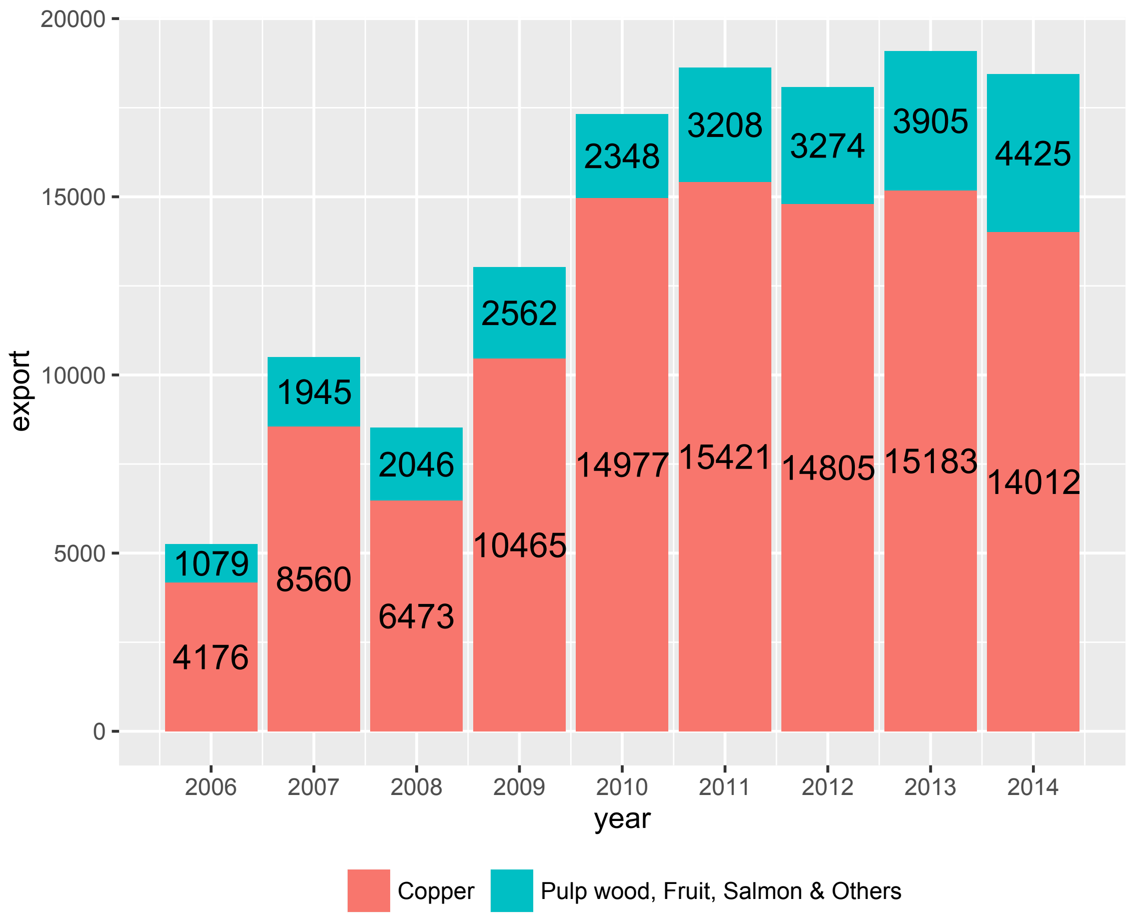
```
charts.data$product <- factor(charts.data$product, levels = c("copper","others"),
  labels = c("Copper ","Pulp wood, Fruit, Salmon & Others"))

p3 <- ggplot() + geom_bar(aes(y = export, x = year, fill = product), data = charts.data, stat="identity
  geom_text(data=charts.data, aes(x = year, y = pos, label = export), size=4),
    show.legend = F) +
  theme(legend.position="bottom", legend.direction="horizontal", legend.title = element_blank())
p3
```



Adjusting x-axis scale

To change the axis tick marks, we use the `scale_x_continuous` and/or `scale_y_continuous` commands.

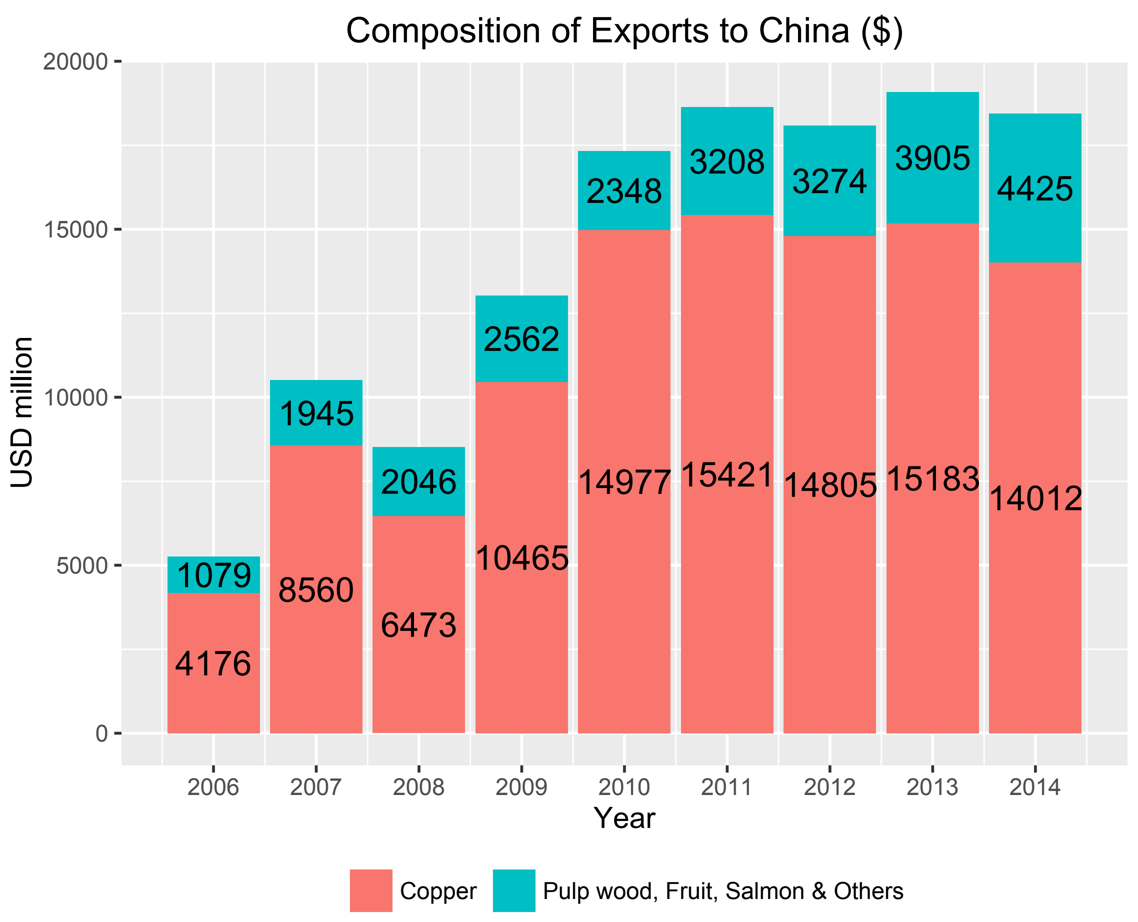
```
p3 <- p3 + scale_x_continuous(breaks=seq(2006,2014,1))
p3
```



Adjusting axis labels & adding title

To add a title, we include the option `ggtitle` and include the name of the graph as a string argument, and to change the axis names we use the `labs` command.

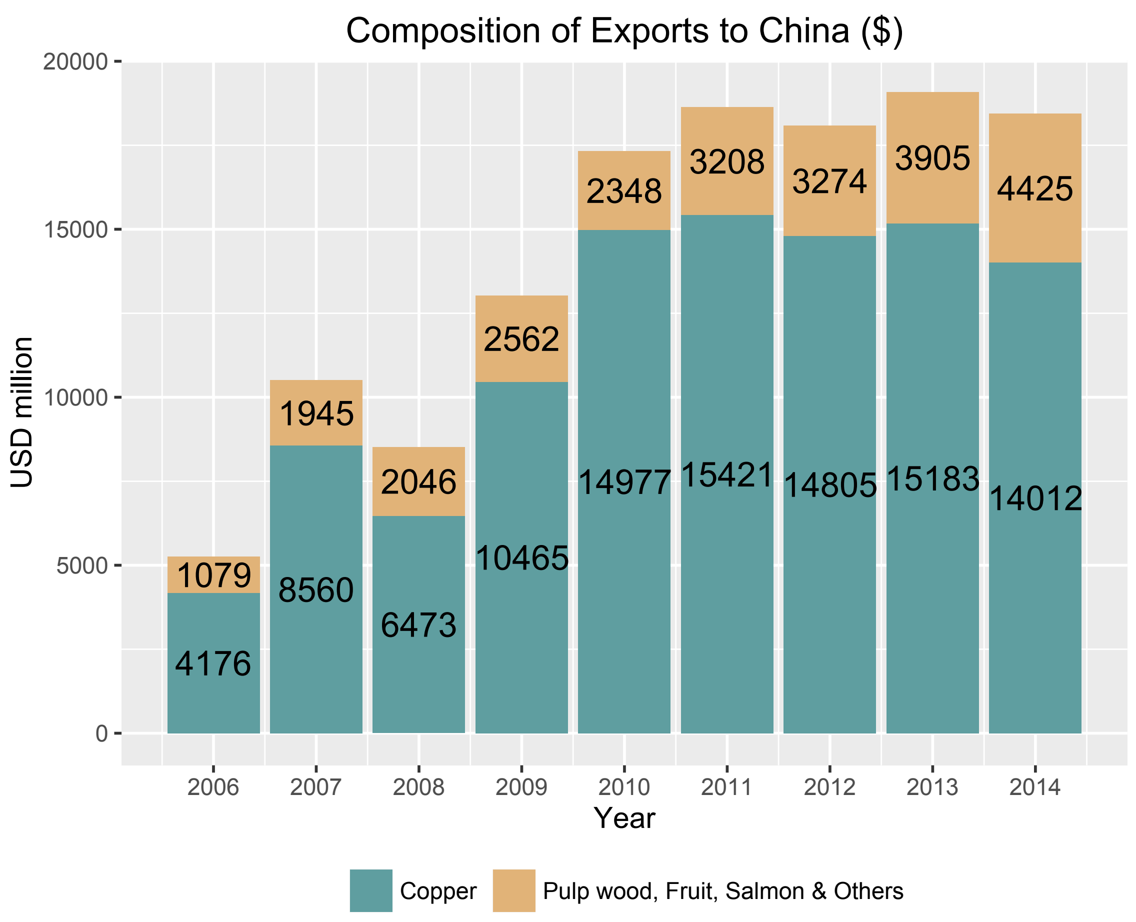
```
p3 <- p3 + ggtitle("Composition of Exports to China ($)") + labs(x="Year", y="USD million")
p3
```



Adjusting color palette

To change the colours, we use the `scale_colour_manual` command. Note that you can reference the specific colours you'd like to use with specific HEX codes. You can also reference colours by name, with the full list of colours recognised by R here.

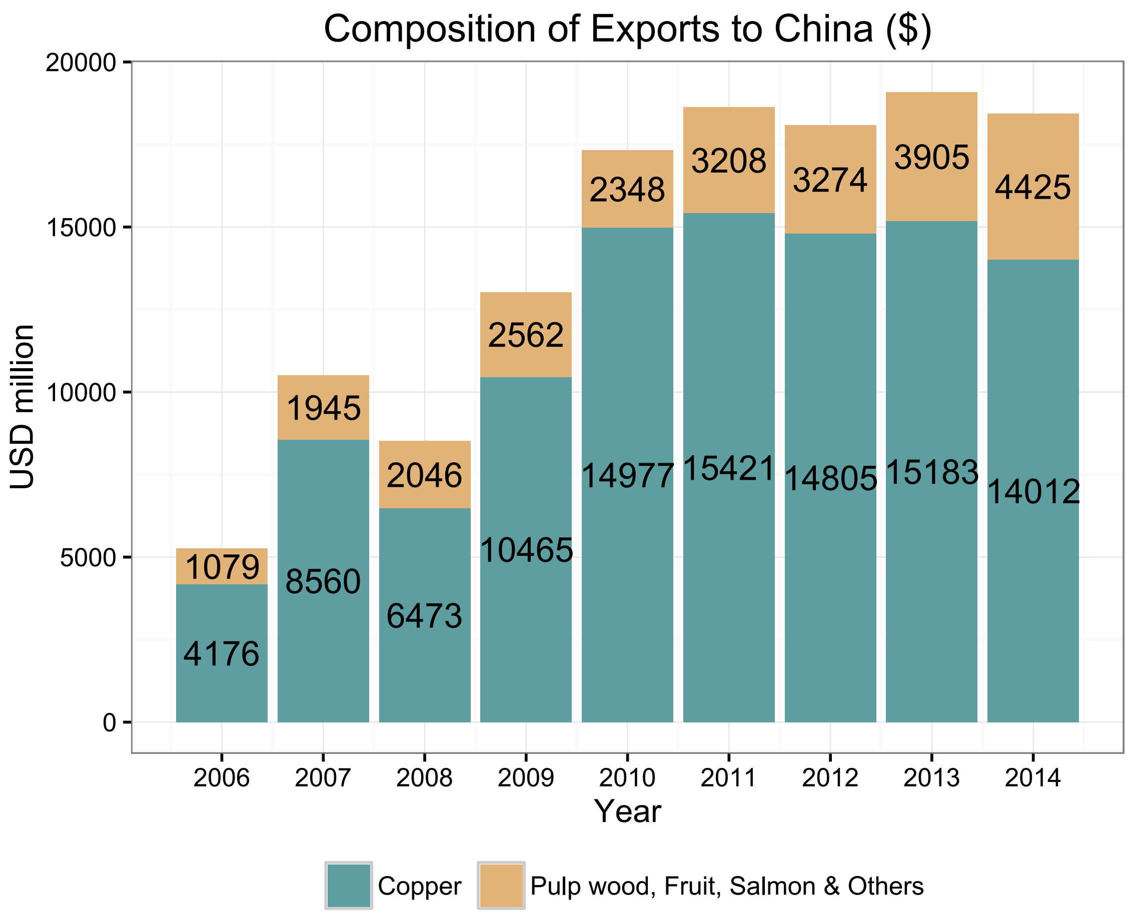
```
fill <- c("#5F9EA0", "#E1B378")
p3 <- p3 + scale_fill_manual(values=fill)
p3
```



Using the white theme

As explained in the previous posts, we can also change the overall look of the graph using themes. We'll start using a simple theme customisation by adding `theme_bw()` after `ggplot()`. As you can see, we can further tweak the graph using the `theme` option, which we've used so far to change the legend.

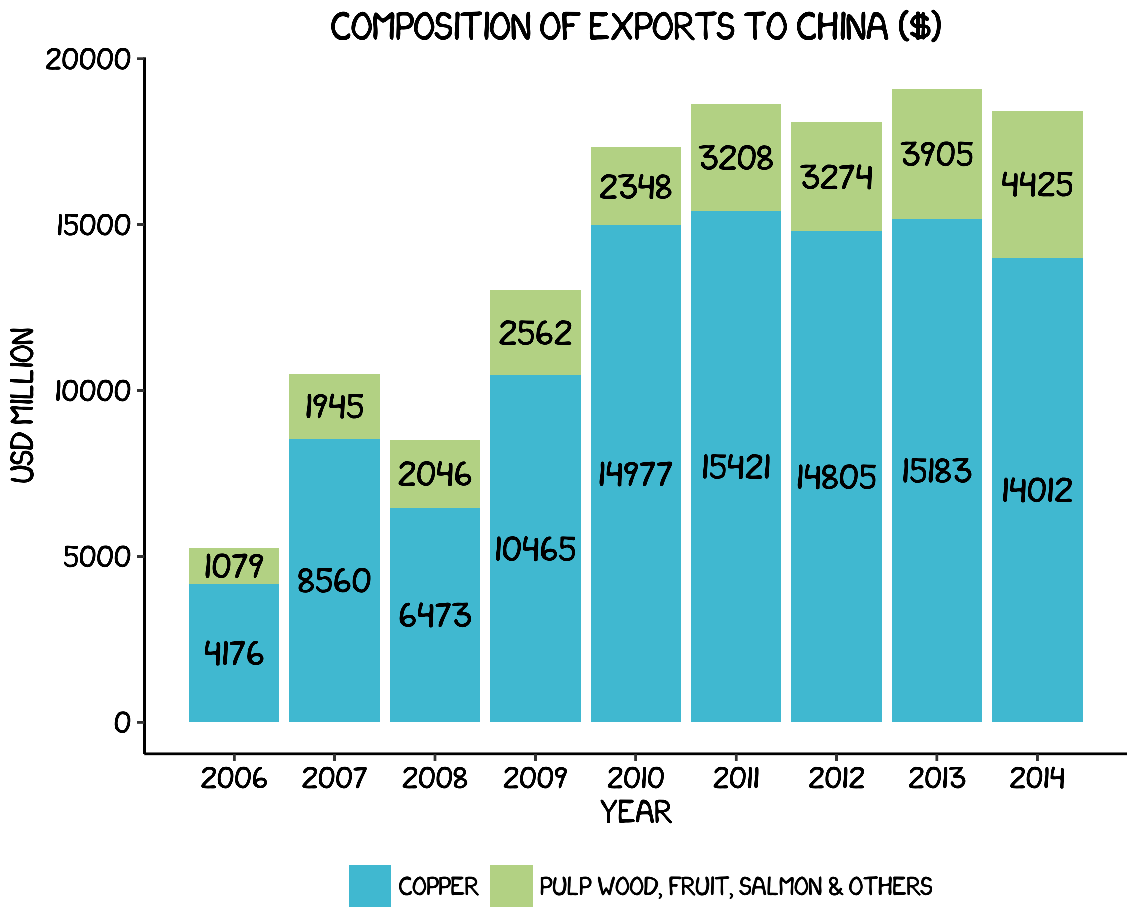
```
p3 <- ggplot() +
  geom_bar(aes(y = export, x = year, fill = product), data = charts.data, stat="identity") +
  geom_text(data=charts.data, aes(x = year, y = pos, label = export, size=4), show.legend = F) +
  scale_x_continuous(breaks=seq(2006,2014,1)) +
  labs(x="Year", y="USD million") +
  ggtitle("Composition of Exports to China ($)") +
  scale_fill_manual(values=fill) +
  theme_bw() +
  theme(legend.position="bottom",
    legend.direction="horizontal",
    legend.title = element_blank())
p3
```



Creating an XKCD style chart

Of course, you may want to create your own themes as well. `ggplot2` allows for a very high degree of customisation, including allowing you to use imported fonts. Below is an example of a theme Mauricio was able to create which mimics the visual style of XKCD. In order to create this chart, you first need to import the XKCD font, and load it into R using the `extrafont` package.

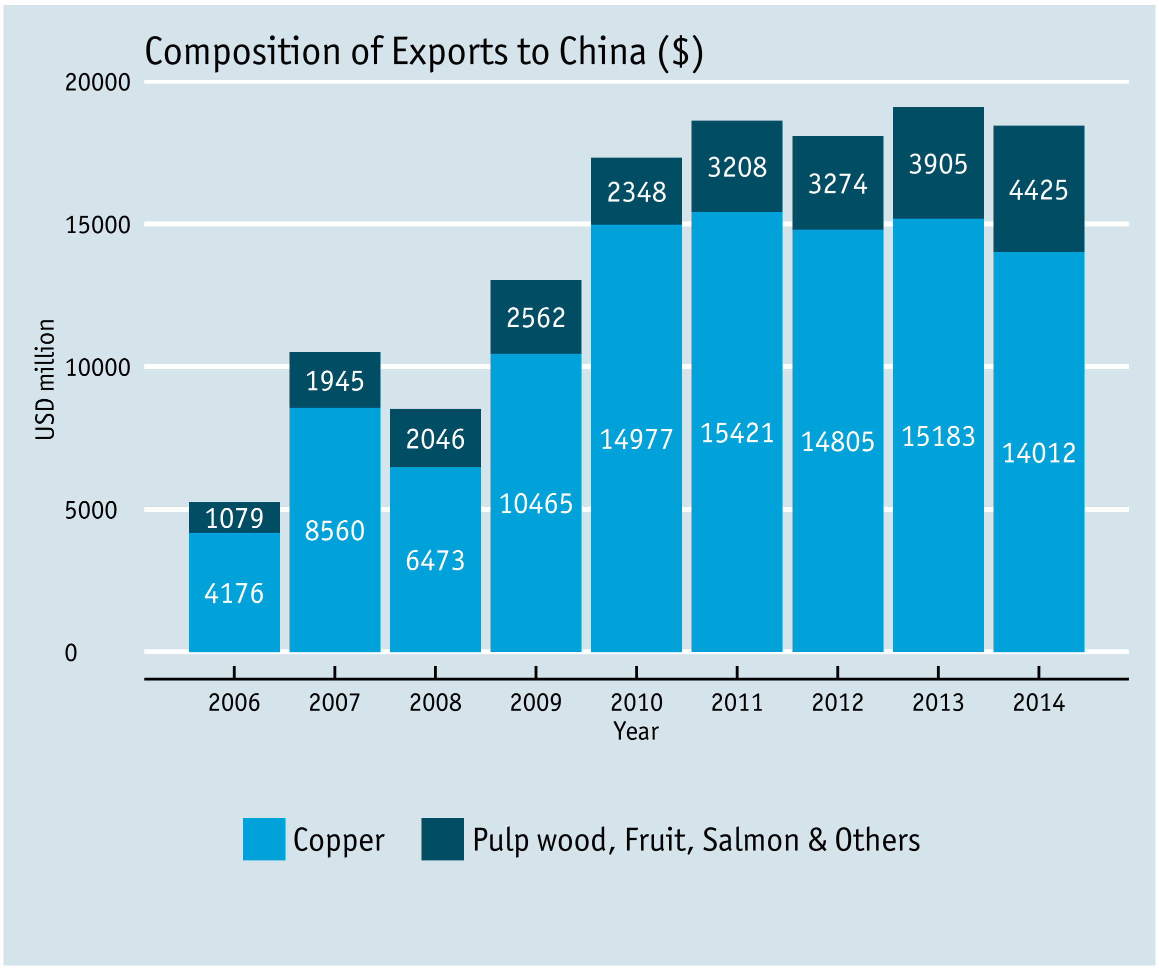
```r
fill <- c("#40b8d0", "#b2d183")

p3 <- ggplot() +
  geom_bar(aes(y = export, x = year, fill = product), data = charts.data, stat="identity") +
  geom_text(data=charts.data, aes(x = year, y = pos, label = export), colour="black",
    family="xkcd-Regular", size = 4, show.legend = F) +
  scale_x_continuous(breaks=seq(2006,2014,1)) +
  labs(x="Year", y="USD million") +
  ggtitle("Composition of Exports to China ($)") +
  scale_fill_manual(values=fill) +
  theme(axis.line.x = element_line(size=.5, colour = "black"),
    axis.line.y = element_line(size=.5, colour = "black"),
    axis.text.x=element_text(colour="black", size = 10),
    axis.text.y=element_text(colour="black", size = 10),
    legend.key=element_rect(fill="white", colour="white"),
    legend.position="bottom", legend.direction="horizontal",
    legend.title = element_blank(),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), panel.border = element_blank(),
    panel.background = element_blank(),
    plot.title=element_text(family="xkcd-Regular"), text=element_text(family="xkcd-Regular"))
p3
```



Using 'The Economist' theme

There are a wider range of pre-built themes available as part of the `ggthemes` package (more information on these here). Below we've applied `theme_economist()`, which approximates graphs in the Economist magazine. It is also important that the font change argument inside `theme` is optional and it's only to obtain a more similar result compared to the original. For an exact result you need 'Officina Sans' which is a commercial font and is available here.

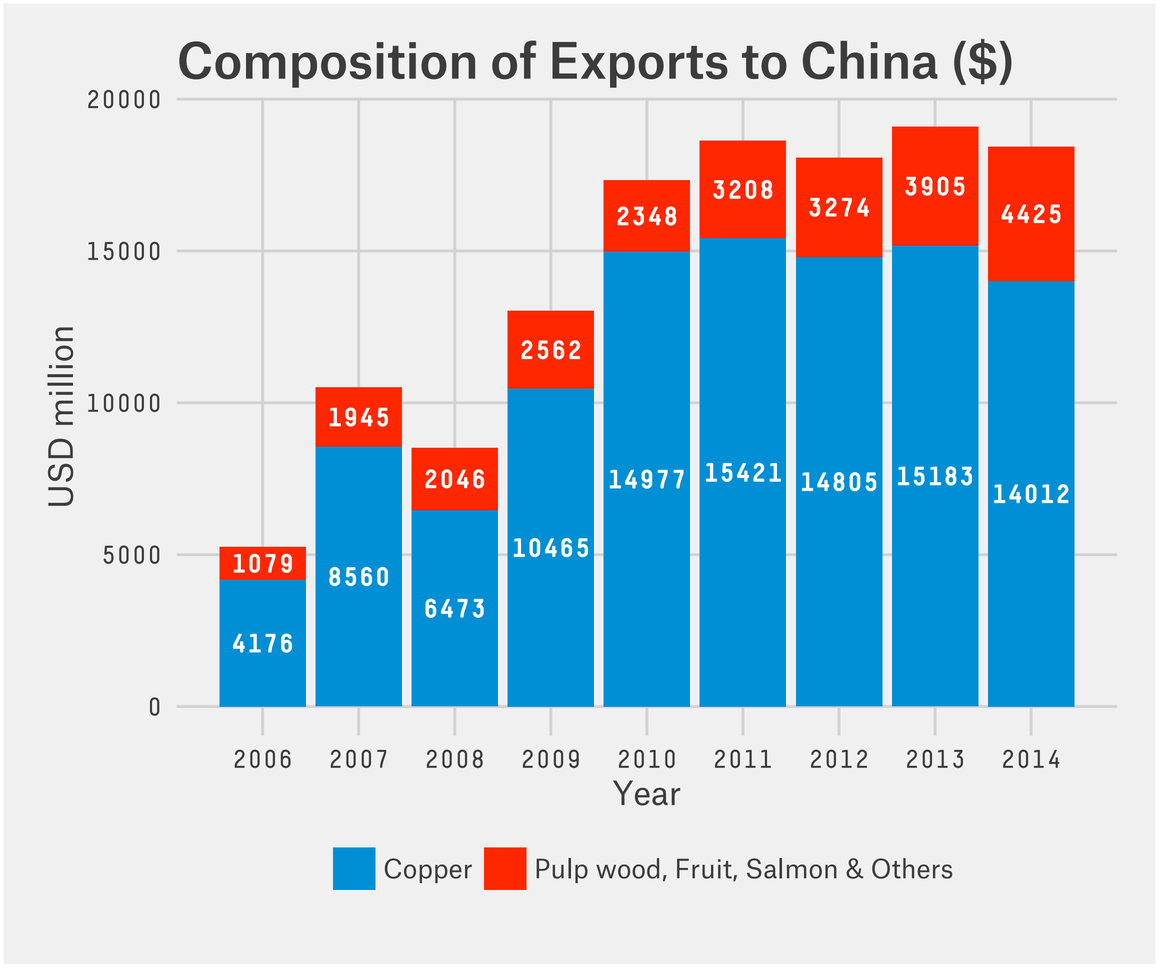
```
p3 <- ggplot() +
  geom_bar(aes(y = export, x = year, fill = product), data = charts.data,
    stat="identity") +
  geom_text(data=charts.data, aes(x = year, y = pos, label = export), colour="white", size = 4, family =
  scale_x_continuous(breaks=seq(2006,2014,1)) +
  labs(x="Year", y="USD million") +
  ggtitle("Composition of Exports to China ($)") +
  theme_economist() + scale_fill_economist() +
  theme(axis.line.x = element_line(size=.5, colour = "black"),
    legend.position="bottom",
    legend.direction="horizontal",
    legend.title = element_blank(),
    plot.title=element_text(family="OfficinaSanITC-Book"),
    text=element_text(family="OfficinaSanITC-Book"))
p3
```



Using 'Five Thirty Eight' theme

Below we've applied `theme_fivethirtyeight()`, which approximates graphs in the nice FiveThirtyEight website. Again, it is also important that the font change is optional and it's only to obtain a more similar result compared to the original. For an exact result you need 'Atlas Grotesk' and 'Decima Mono Pro' which are commercial font and are available here and here.

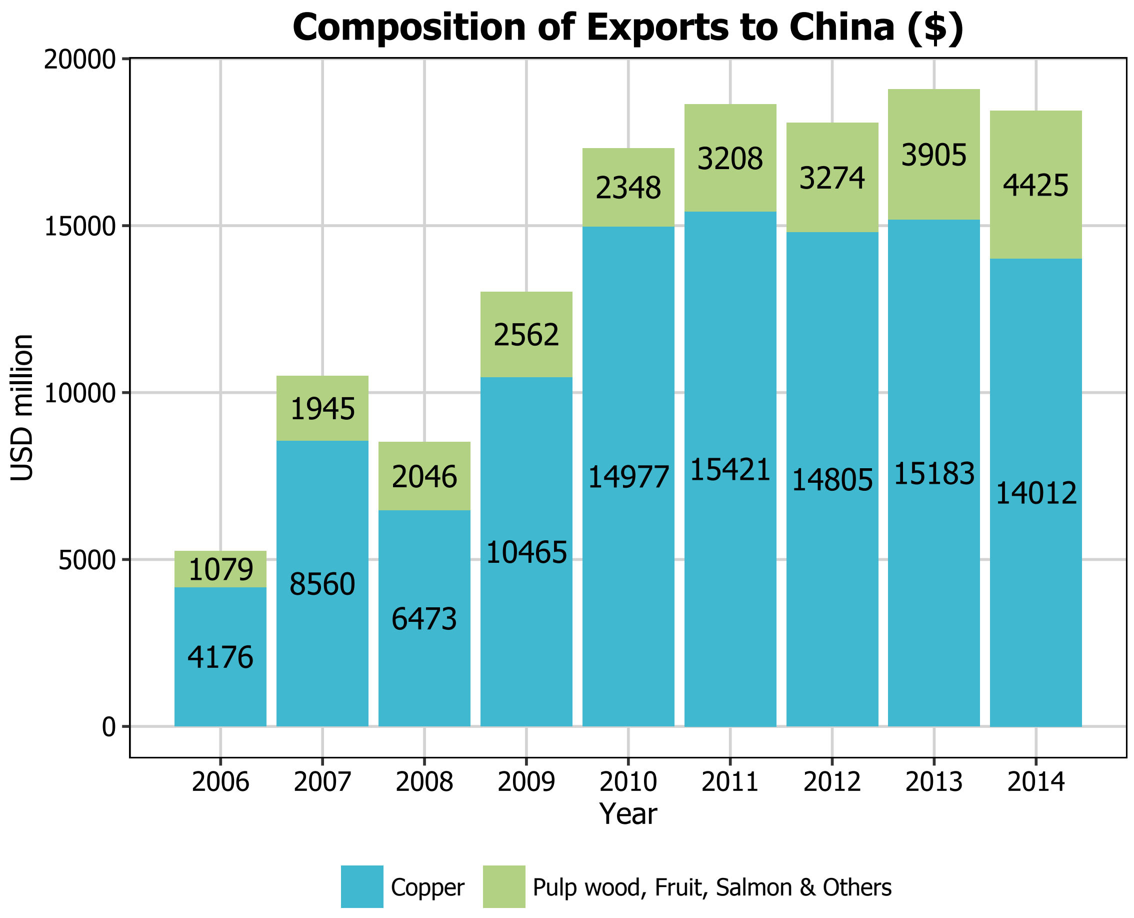
```
p3 <- ggplot() +
  geom_bar(aes(y = export, x = year, fill = product), data = charts.data, stat="identity") +
  geom_text(data=charts.data, aes(x = year, y = pos, label = export), colour="white", size = 3.5, famil
  scale_x_continuous(breaks=seq(2006,2014,1)) +
  labs(x="Year", y="USD million") +
  ggtitle("Composition of Exports to China ($)") +
  theme_fivethirtyeight() + scale_fill_fivethirtyeight() +
  theme(axis.title = element_text(family="Atlas Grotesk Regular"),
    legend.position="bottom", legend.direction="horizontal",
    legend.title=element_blank(),
    plot.title=element_text(family="Atlas Grotesk Medium"),
    legend.text=element_text(family="Atlas Grotesk Regular"),
    text=element_text(family="DecimaMonoPro"))
p3
```



Creating your own theme

As before, you can modify your plots a lot as `ggplot2` allows many customisations. Here we present our original result shown at the top of page.

```r
fill <- c("#40b8d0", "#b2d183")

p3 <- ggplot() +
  geom_bar(aes(y = export, x = year, fill = product), data = charts.data, stat="identity") +
  geom_text(data=charts.data, aes(x = year, y = pos, label = export), colour="black",
    family="Tahoma", size = 4, show.legend = F) +
  scale_x_continuous(breaks=seq(2006,2014,1)) +
  labs(x="Year", y="USD million") +
  ggtitle("Composition of Exports to China ($)") +
  scale_fill_manual(values=fill) +
  theme(panel.border = element_rect(colour = "black", fill=NA, size=.5),
    axis.text.x=element_text(colour="black", size = 10),
    axis.text.y=element_text(colour="black", size = 10),
    legend.key=element_rect(fill="white", colour="white"),
    legend.position="bottom", legend.direction="horizontal",
    legend.title = element_blank(),
    panel.grid.major = element_line(colour = "#d3d3d3"),
    panel.grid.minor = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank(),
    plot.title = element_text(size = 14, family = "Tahoma", face = "bold"),
    text=element_text(family="Tahoma"))
p3
```



Composition of Exports to China ($)
USD million
20000
15000
10000
5000
0
1079
4176
1945
8560
2046
6473
2562
10465
2348
14977
3208
15421
3274
14805
3905
15183
4425
14012
2006
2007
2008
2009
2010
2011
2012
2013
2014
Year
Copper
Pulp wood, Fruit, Salmon & Others