# Exploring ggplot

*Paul Brennan*

*15 May 2017*

## Plotting your data

Plotting is a key aspect of data analysis. It is valuable during two distinct phases: data exploration (at the beginning of the analysis cycle); and data presentation (often at the end of the analysis cycle).

Base R will generate some interesting plots and it's worth trying the plot() function to see what happens. Another useful function for multidimensional data is pairs() which allows multiple comparisons.

Mostly, I plan to focus on using the ggplot2 library - a key author is Hadley Wickham. There is very good online documentation available here http://ggplot2.tidyverse.org/ and a book available on github https://github.com/hadley/ggplot2-book

## Making a bar chart

I would like to start making a bar chart. I have downloaded some data about the Namibia population - the percentage of people living in an Urban and Rural locations. This was downloaded from http://namibia.opendataforafrica.org

## Step 1: important the data

The data is comma separated so can be imported using the read.csv() function.

```
## making a nice bar chart using ggplot
# data available from http://namibia.opendataforafrica.org/NDPNPHCV2017/namibia-population-and-housing-
# namibia Urban Rural population
# this is an interesting option for drawing.

# download the data
library(RCurl)
```

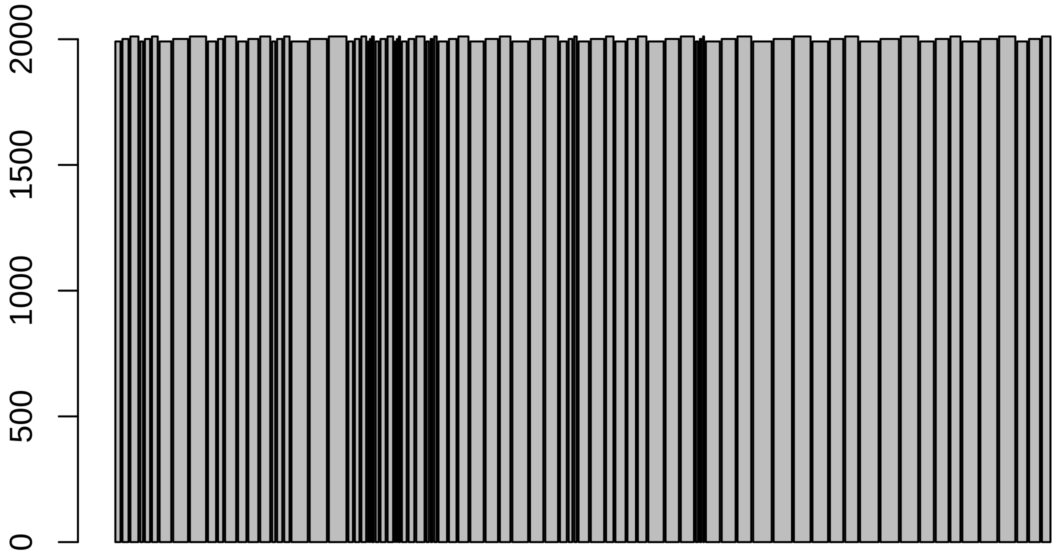
```
## Loading required package: bitops
```

```
x <- getURL("https://raw.githubusercontent.com/brennanpincardiff/learnR/master/data/Namibia_region_urba
data <- read.csv(text = x)
View(data) # have a look at the data
str(data)  # have a look at the structure
```

```
## 'data.frame':    84 obs. of  7 variables:
##  $ region   : Factor w/ 14 levels "Erongo","Hardap",..: 7 7 7 14 14 14 1 1 1 2 ...
##  $ variable : Factor w/ 1 level "Percentage of Total population in Urban/Rural": 1 1 1 1 1 1 1 1 1 1
##  $ sex      : Factor w/ 1 level "Total": 1 1 1 1 1 1 1 1 1 1 ...
##  $ residence: Factor w/ 2 levels "Rural","Urban": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Unit     : Factor w/ 1 level "%": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Date     : int  1991 2001 2011 1991 2001 2011 1991 2001 2011 1991 ...
##  $ Value    : int  28 33 43 15 28 31 63 80 87 44 ...
```

### Make a bar chart with base R

```
barplot(data$Date, data$Value)
```



Because of the structure and complexity of the data, this plot is not very useful.

### Let's use ggplot2

I recommend the use of library("ggplot2") to generate more useful plots.

```
library("ggplot2")
```

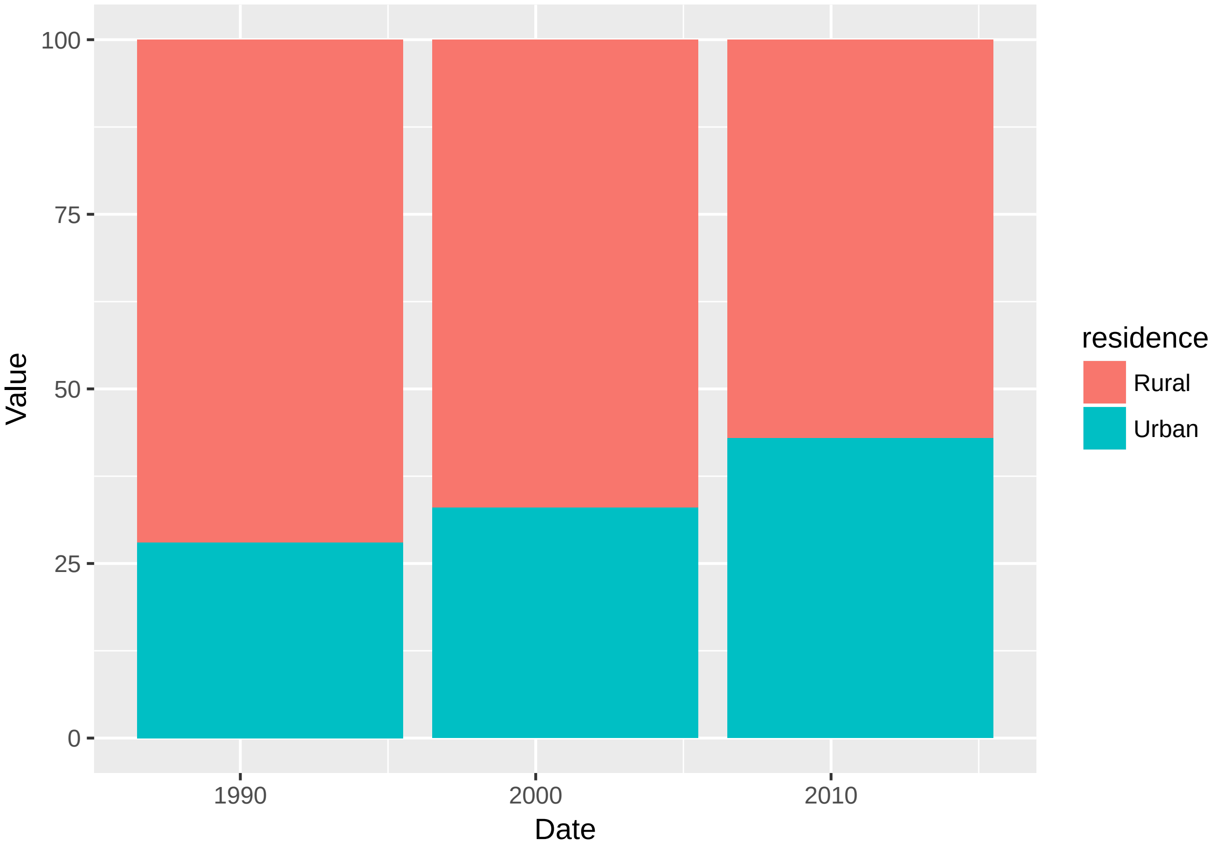
The ggplot() function requires you to give it:

1. data,
2. the aesthetics - in this case x, y and fill
3. and a geometry - in this case geom_bar()

Below is the code Note:

1. We have subset the data
2. Our x value is Date
3. Our y value is Value (a percentage)
4. Our fill is residence
5. The plus (+) sign adds…
6. geom_bar( )
7. Which has one arguement for stat (statistic)

```
# graph Namibia over the three years we have data for...
ggplot(data = data[data$region=="Namibia",],
       aes( x = Date,
            y = Value,
            fill = residence)) +
  geom_bar(stat = "identity")
```



Note: Date at the bottom - doesn't look quite correct The labels are slightly off centre to bars

```
unique(data$Date)
```

```
## [1] 1991 2001 2011
```

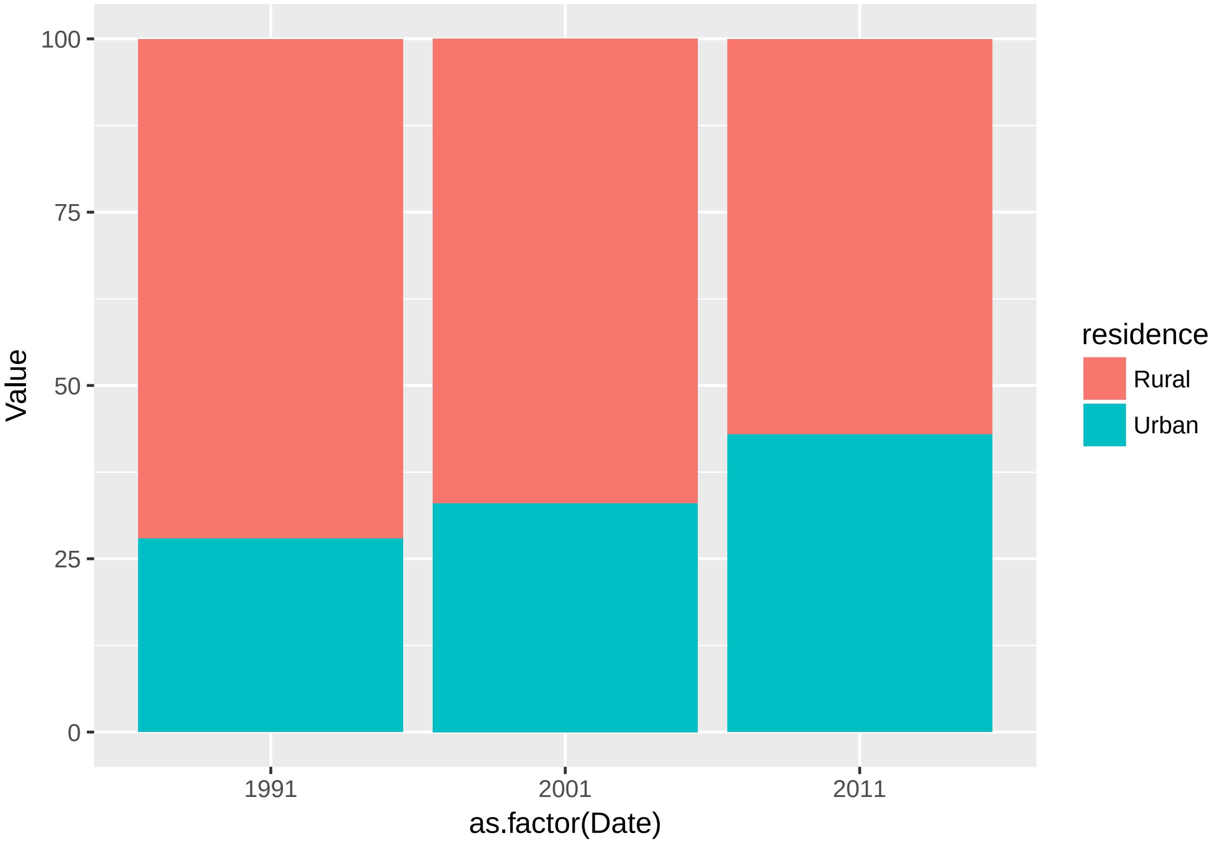
```
# three years are 1991, 2001, 2011

# R is treating these dates as a integer
str(data$Date)
```

```
##  int [1:84] 1991 2001 2011 1991 2001 2011 1991 2001 2011 1991 ...
```

This data will plot better if we convert Date to factor before we plot

```
ggplot(data = data[data$region=="Namibia",],
       aes( x = as.factor(Date),
            y = Value,
            fill = residence)) +
  geom_bar(stat = "identity")
```

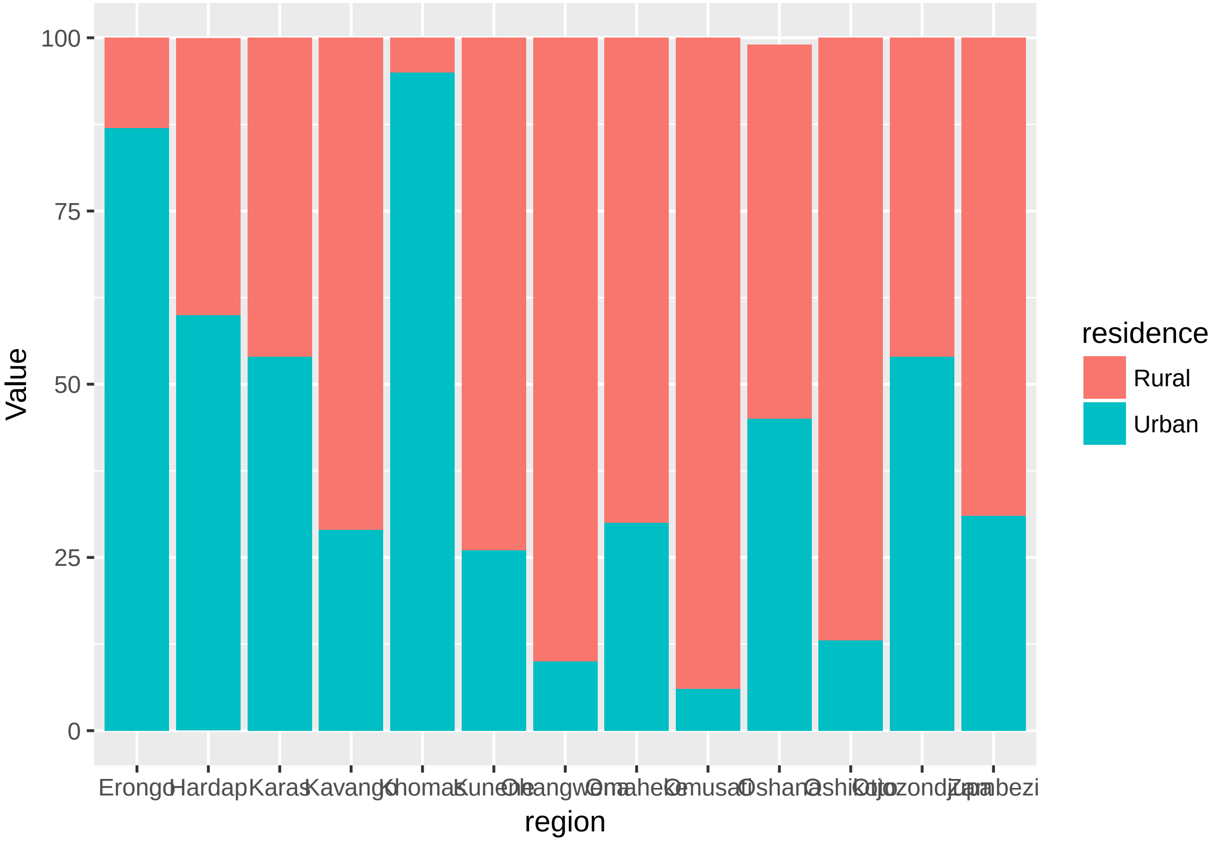


```
# now the labels are in the middle of the bars
```

## Plotting regional data

So that is our basic bar plot using geom_bar( ). We want to extend this by exploring regional variation. We subset the data to exclude the whole country values. The exclamation mark "!" does this.

```
# subset the data to exclude whole country
data_regions <- data[!data$region=="Namibia",]
# focus on 2011 data
ggplot(data = data_regions[data_regions$Date == 2011,],
       aes(x = region,
           y = Value,
           fill = residence)) +
  geom_bar(stat = "identity")
```



```
# zoom to see the y-labels more clearly
```

The plots not perfect but we can begin to see how it works.

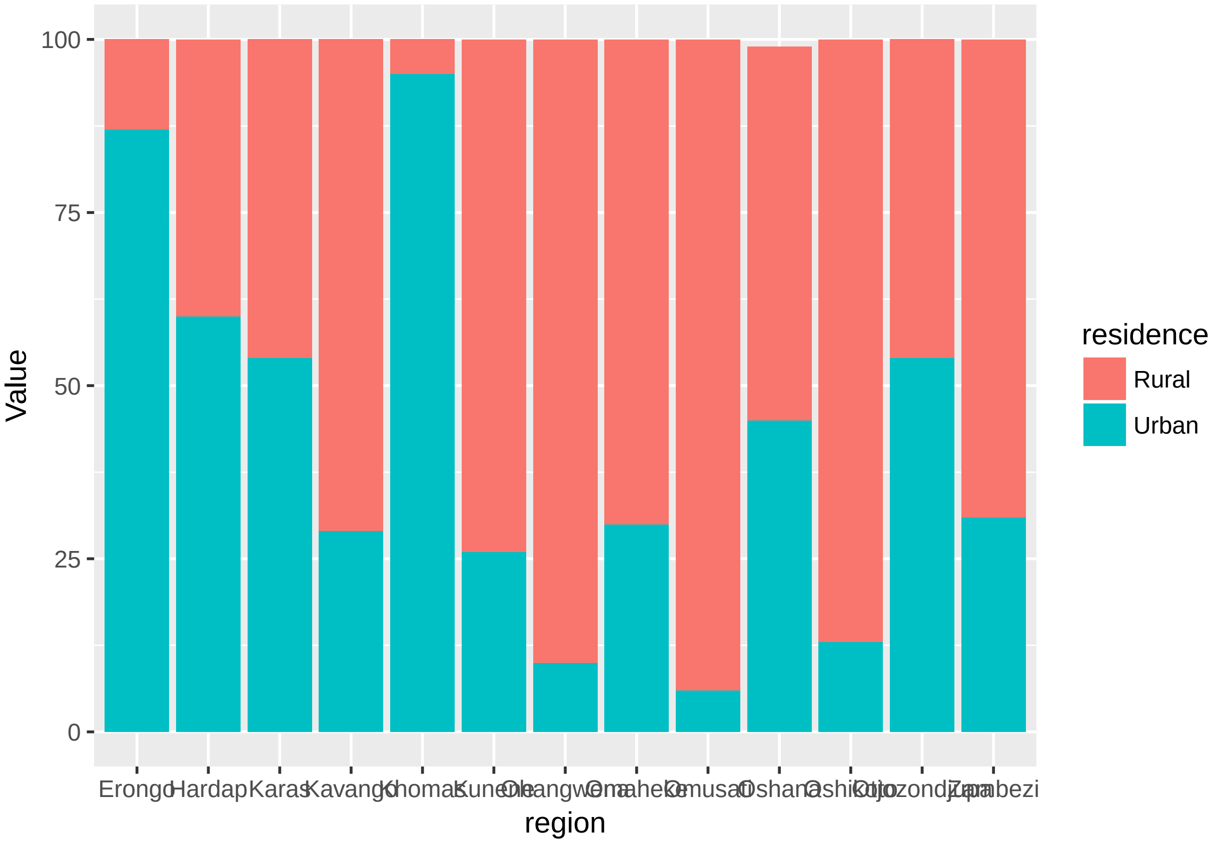
### Changing a theme and add a title

First create an object with the plot in it.

```r
p <- ggplot(data = data_regions[data_regions$Date == 2011,],
            aes(x = region,
                y = Value,
                fill = residence)) +
  geom_bar(stat = "identity")
```

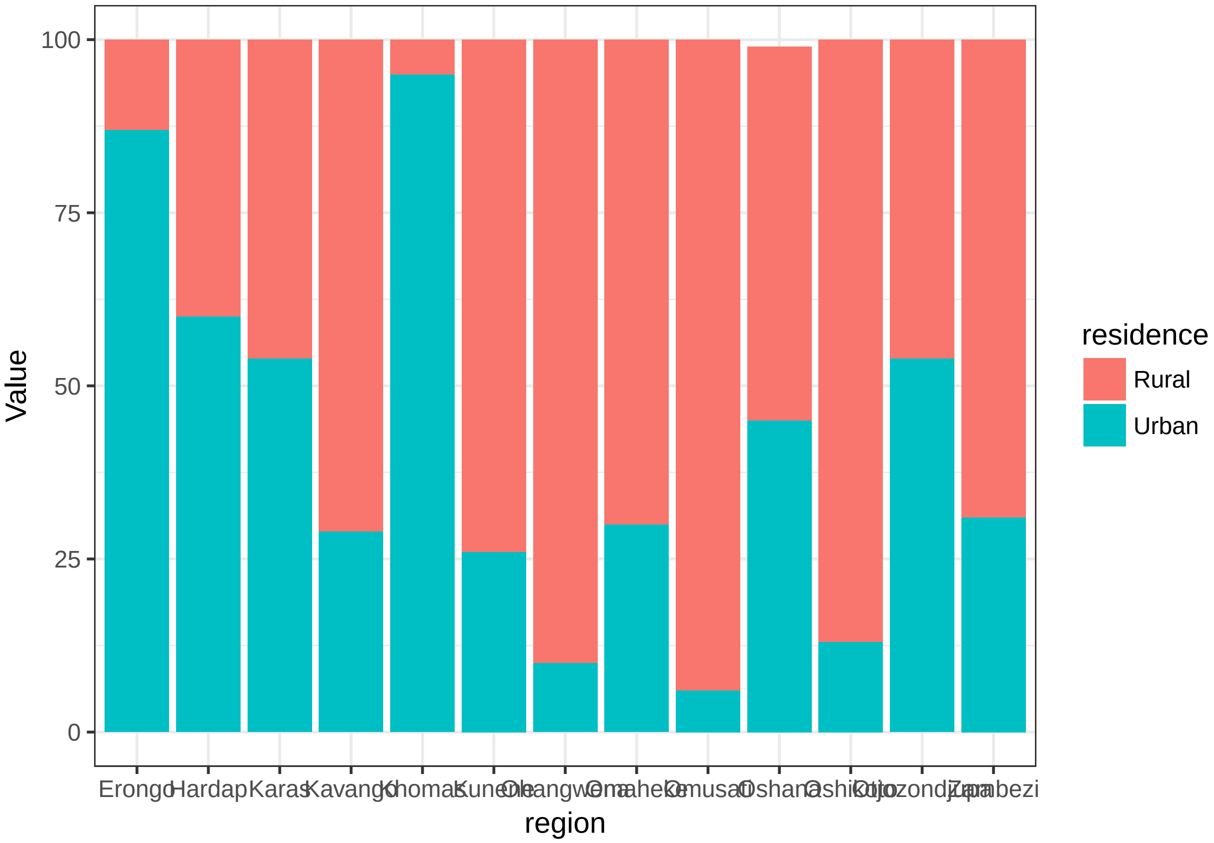
Look at the Global Environment and the object “p” is there. It’s a “List of 9” If you press the blue arrow beside the object “p”, it shows some of the features of the object. Note theme: list() - so there is no theme. To show the object type p

```r
p
```



We can alter the display of this object by adding a plus sign “+”

```
# show a different theme
p + theme_bw()    # shows but does NOT modify the object
```



It’s important to note that this doesn’t alter the object and so represents a temporary change to the display of the object.

To modify the object, this is the syntax:

```
# modify the object with the bw theme
p <- p + theme_bw()
```

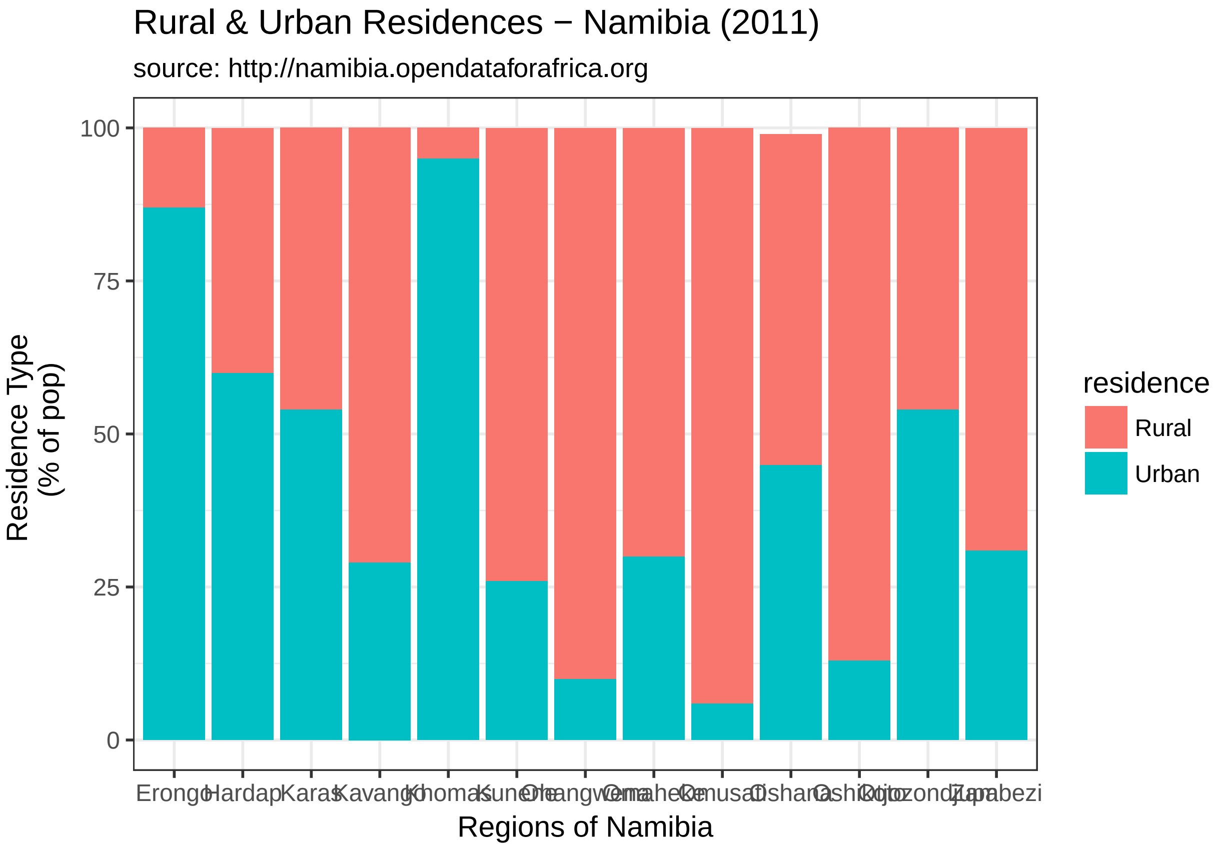
This modify the object but does NOT display it. Check out the theme part of the object in the Global Environment. theme is now a List of 57!

### Add Titles

A good plot has titles so let's add those.

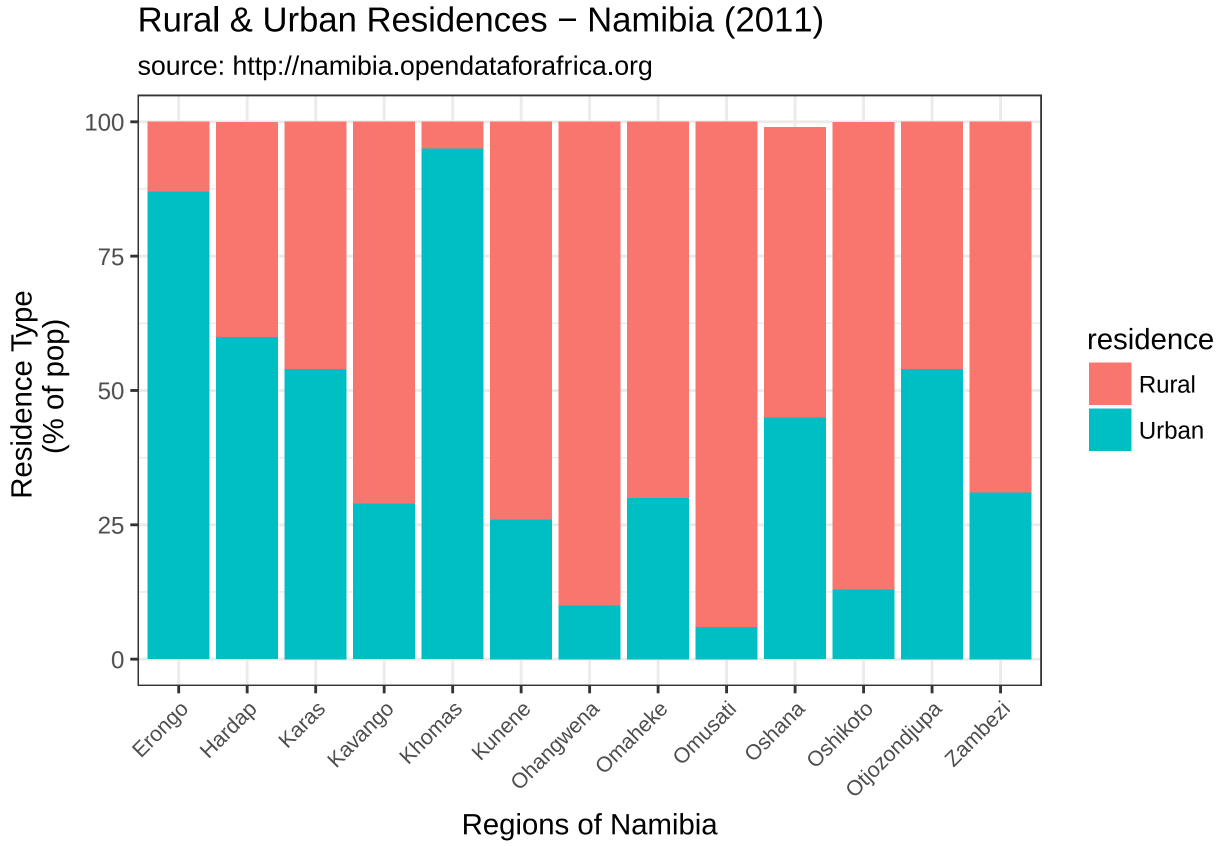
```r
# add titles
p <- p + labs(x = "Regions of Namibia",          # label x-axis
         y = "Residence Type \n (% of pop)",  # label y-axis
         title = "Rural & Urban Residences - Namibia (2011)",
         subtitle = "source: http://namibia.opendataforafrica.org")

p  # show the object
```



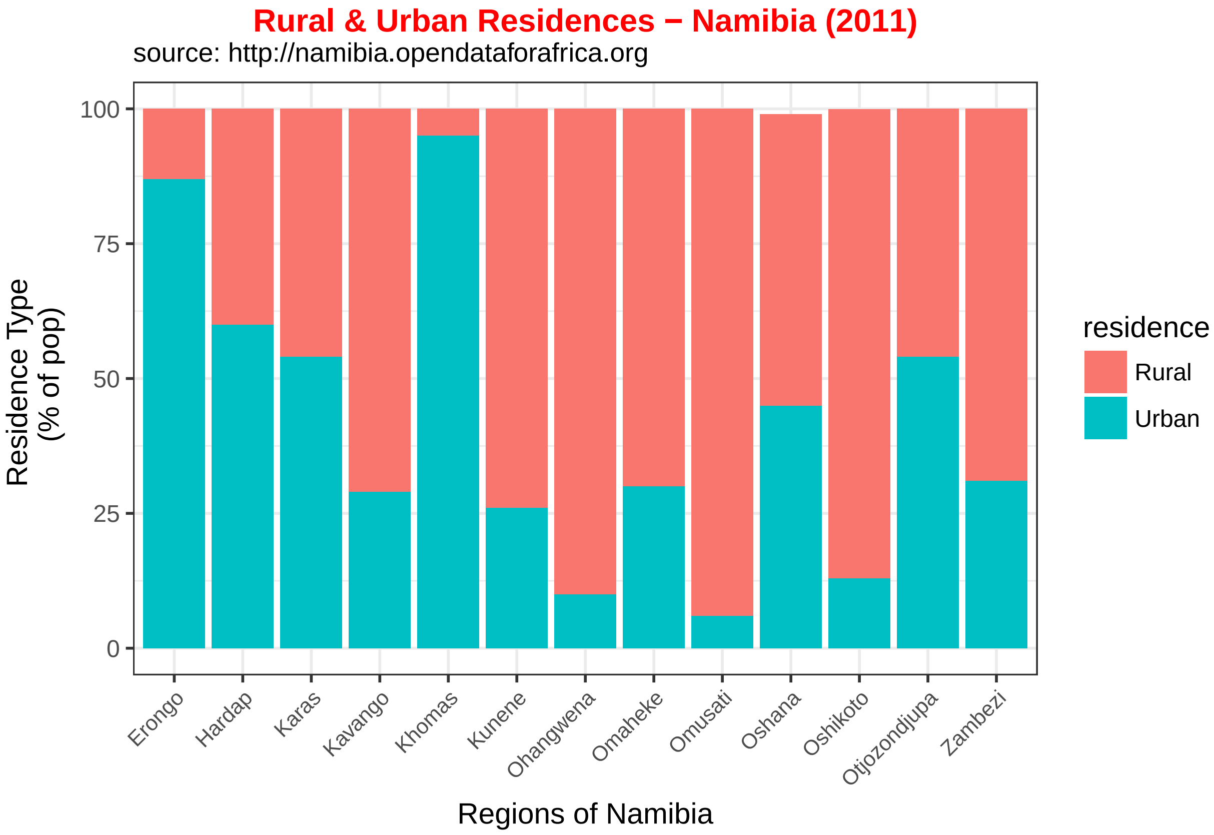
### Change orientation of text on x-axis

```
# http://stackoverflow.com/questions/11724311/how-to-add-a-ggplot2-subtitle-with-different-size-and-col
# customise the format of the text
# first change the angle of the text on the x axis
p <- p + theme(axis.text.x=element_text(size=8,
                                        angle=45,
                                        hjust=1))
p  # show the object
```



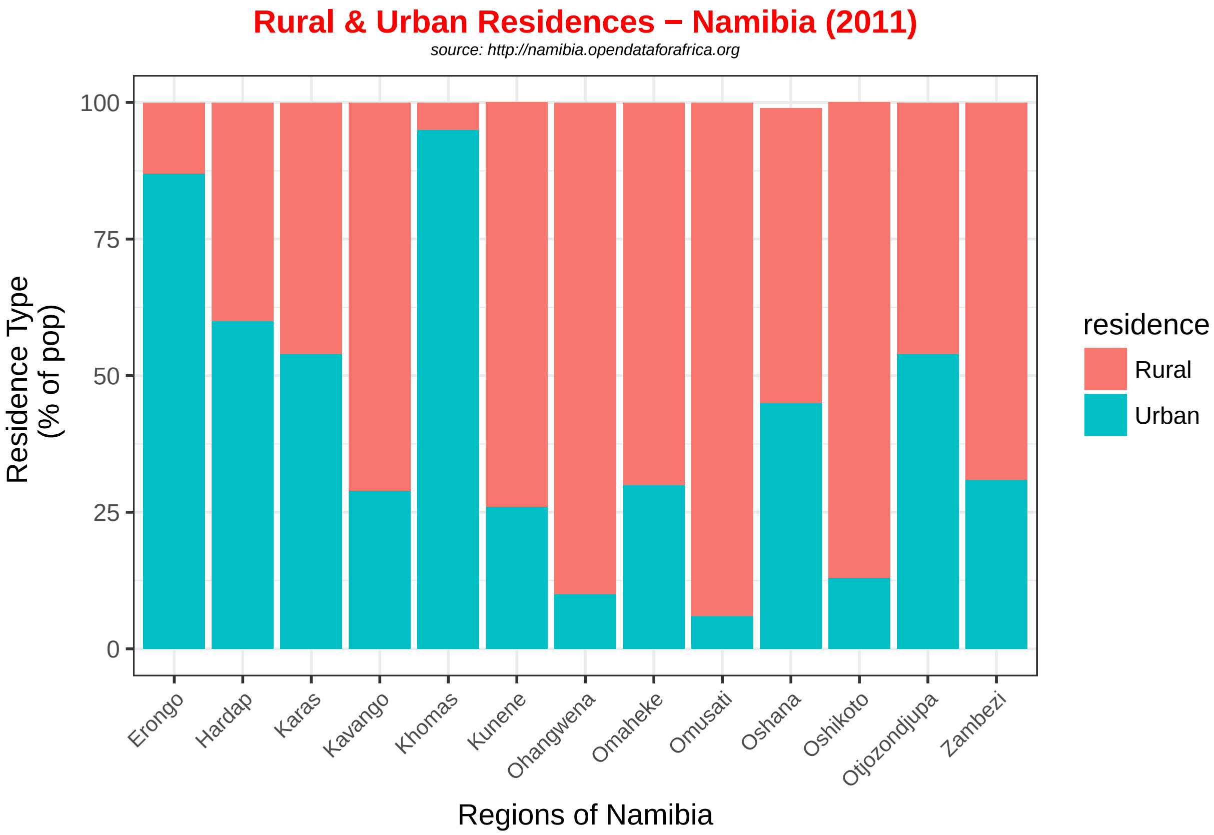
### Customise the title with colour and bold and position

```
p <- p + theme(plot.title=element_text(size=12,
                                        hjust=0.5,
                                        face="bold",
                                        colour="red",
                                        vjust=-1))
p  # show the object
```



### Customise the subtitle which has the source of the data.

```
p <- p + theme(plot.subtitle=element_text(size=6, hjust=0.5, face="italic", color="black"))
p # show the plot
```

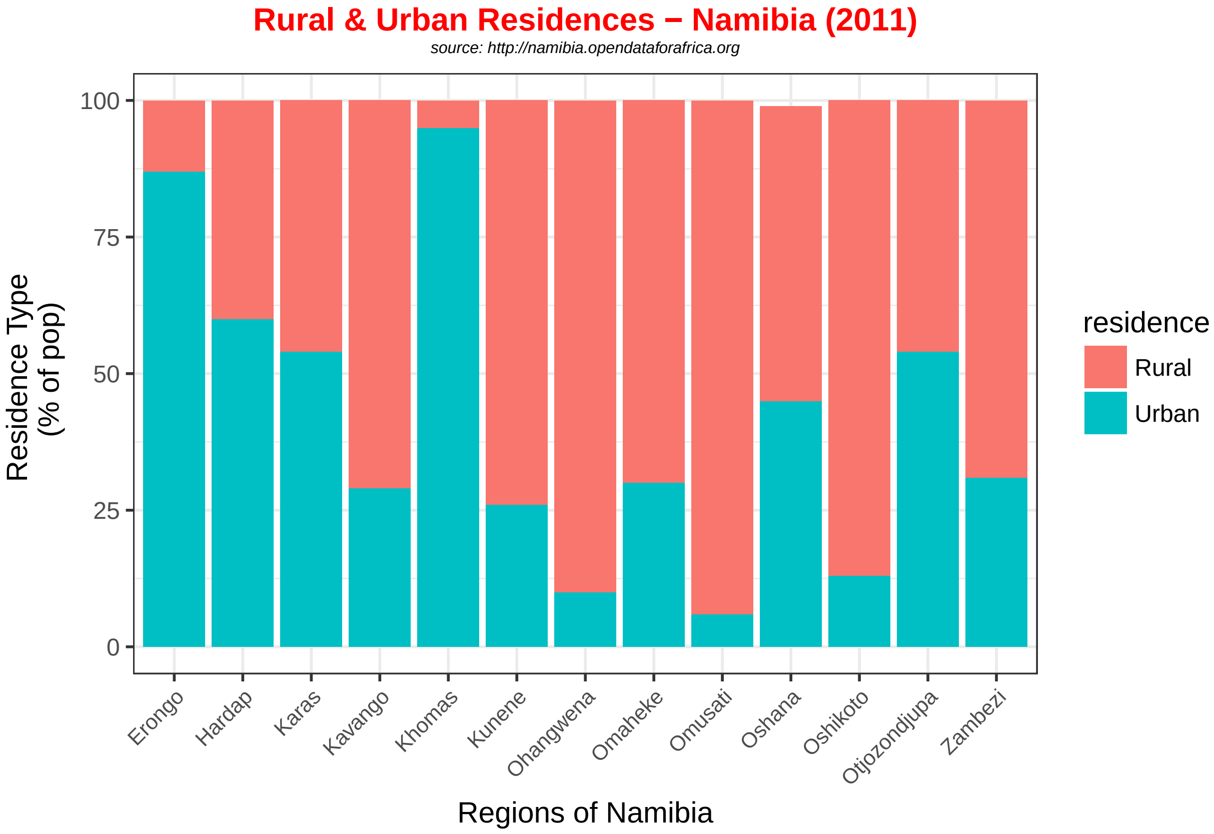


### Export and saving your plot

Happy with your plot? You can "Export" it using the button in Plots or you can add the ggsave() function and give it a file name.

```
# export or save the plot
p + ggsave("Namibia_Regions_Residences_2011.pdf")
```

```
## Saving 6.5 x 4.5 in image
```



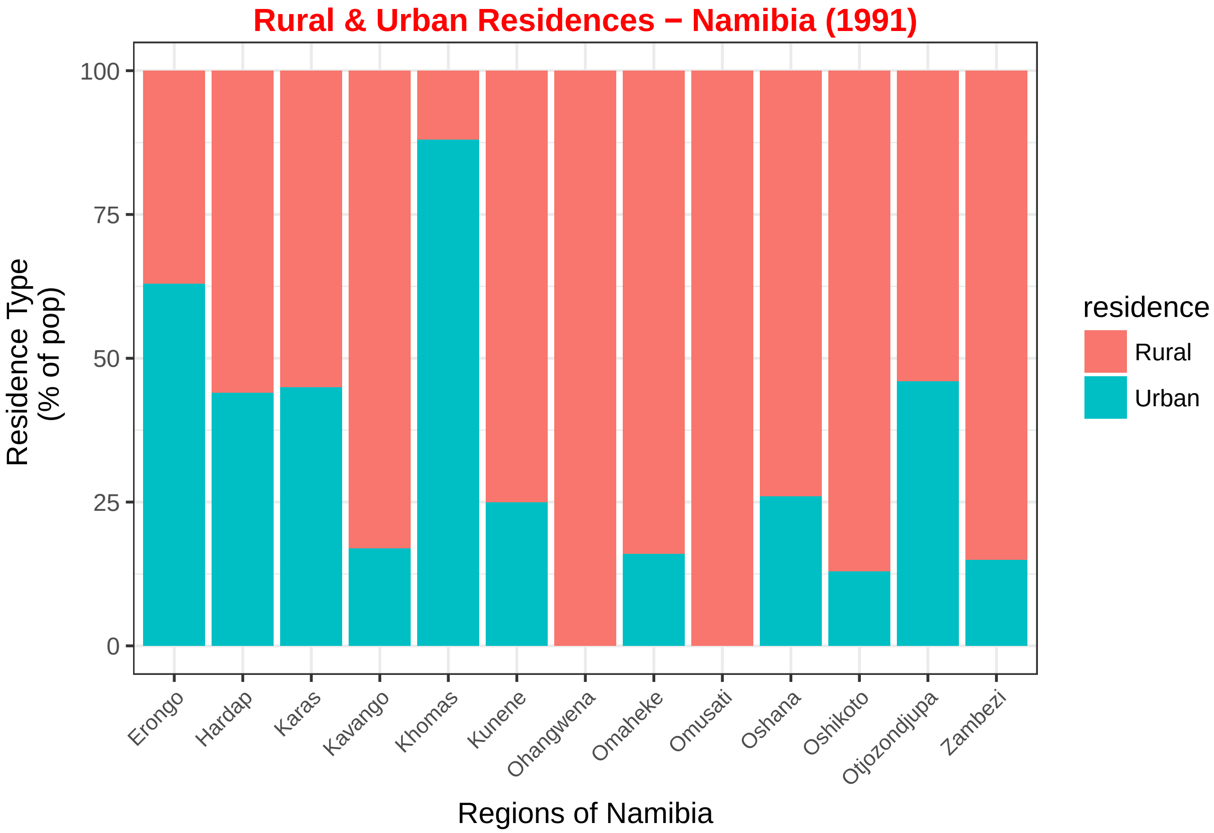
### Re-using your plot with new data.

One of the things, I most like about R and ggplot2 is the ability to automate and force in different data into plots that like. Our data has three years so let's see if I can make the same plot but with different data - for example 1991 The %+% command does this.
Here is the syntax:

```
p1991 <- p %+% data_regions[data_regions$Date == 1991,]
```

We have created a new object. Look at the object and we find $ Date is now 1991 - so that's good. We also want to change the title.

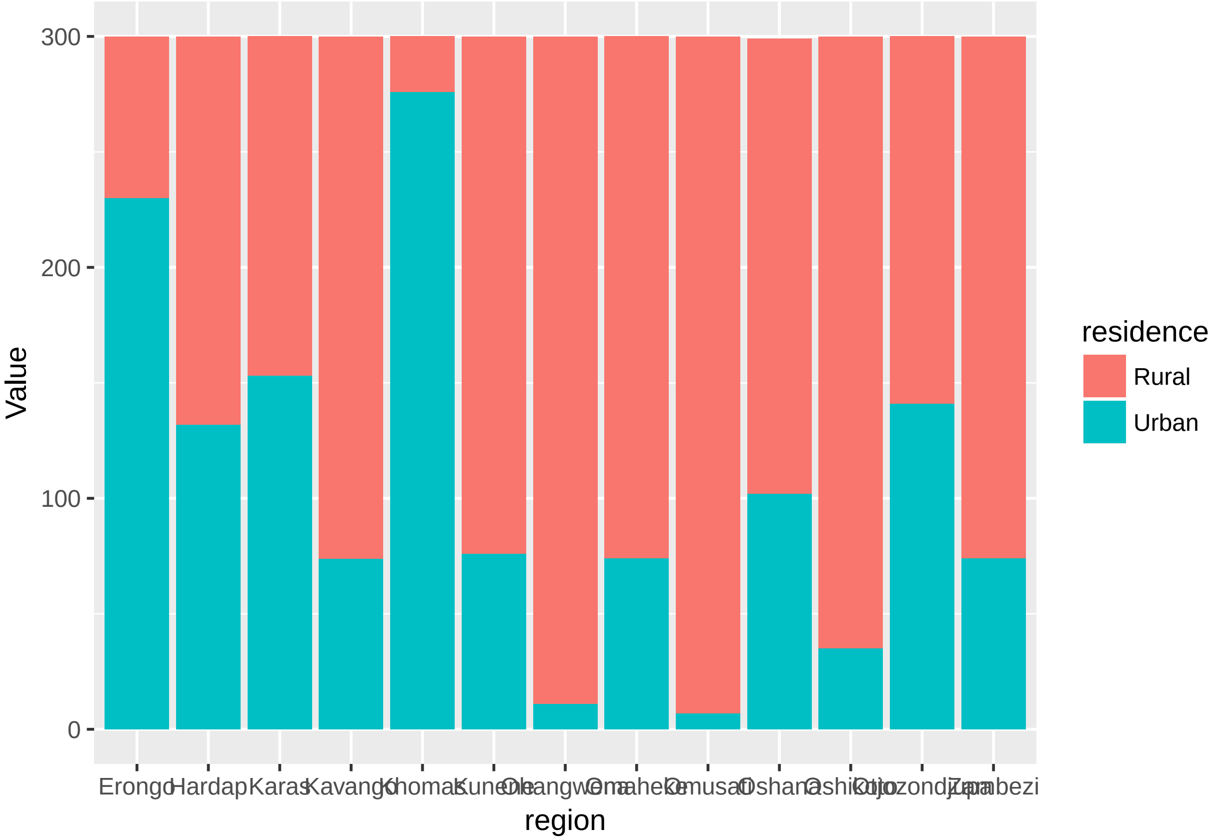
```
p1991 <- p1991 + ggtitle("Rural & Urban Residences - Namibia (1991)")
p1991 # show the object
```



### Using faceting to show multiple plots.

Our data has three years within it. It's possible to plot all three together. First step is to make an object with all the data.

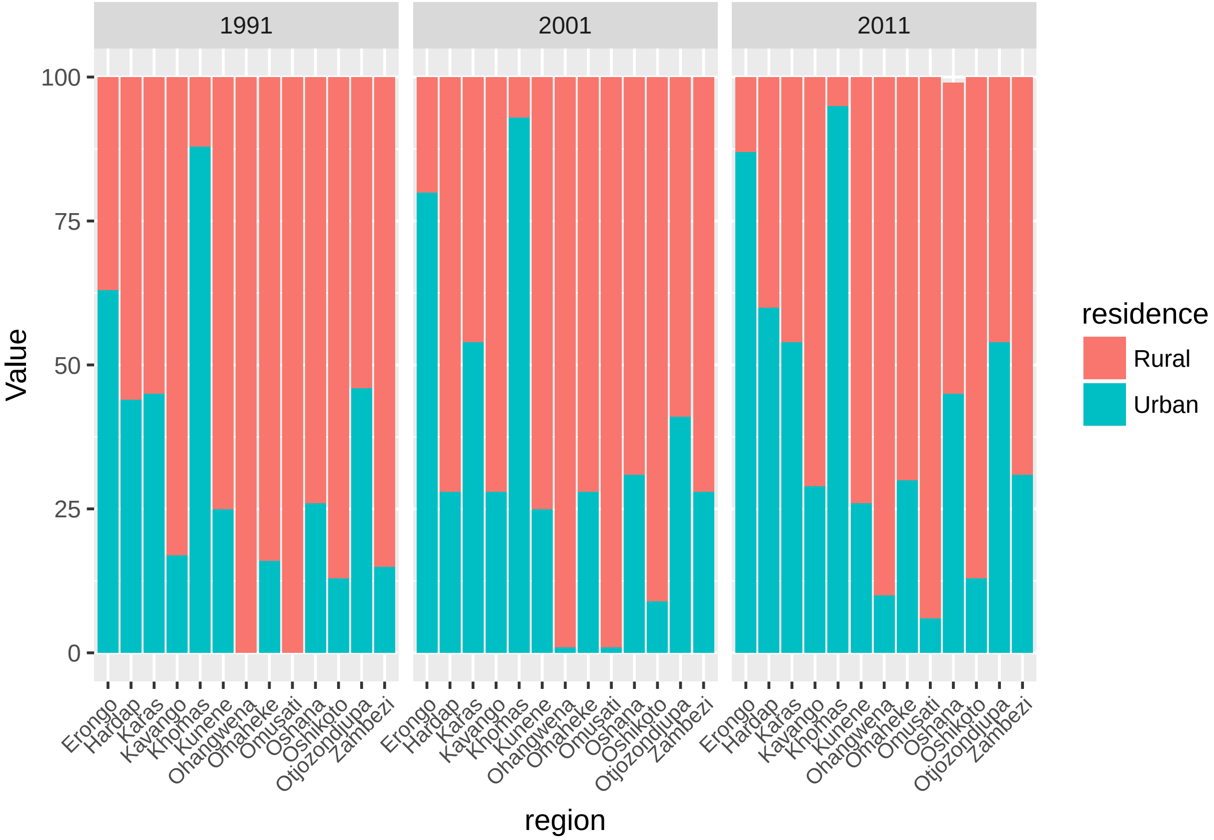
```
# make an object with all the data
f <- ggplot(data = data_regions,
       aes(x = region,
           y = Value,
           fill = residence)) +
  geom_bar(stat = "identity")
f # show the object
```



```
# the geom_bar adds all the values together
# see the y axis goes up to 300.
```

The we use facet_wrap() to separate them again.

```
# use facet_wrap to separate them again
f + facet_wrap(~Date) +
  theme(axis.text.x=element_text(size=8,
                                 angle=45,
                                 hjust=1))
```

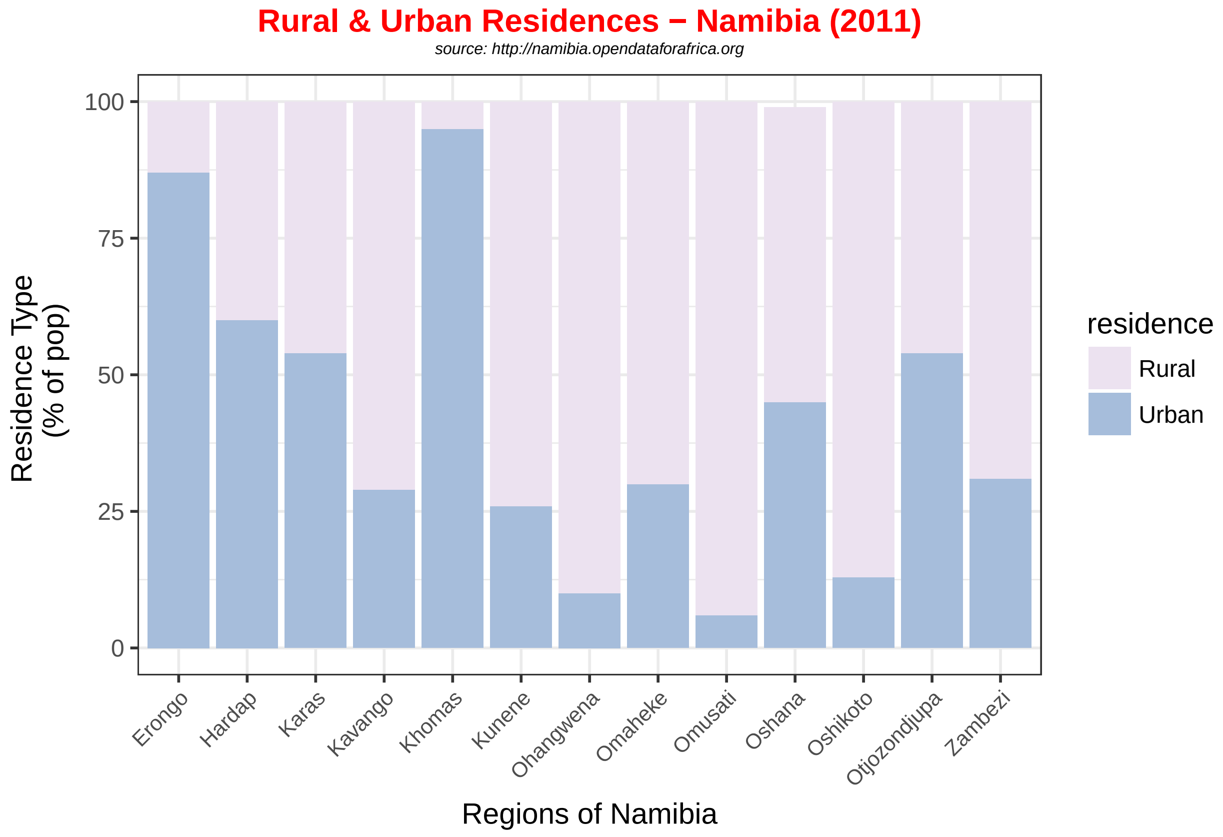


```
# remember this will just display the object
```

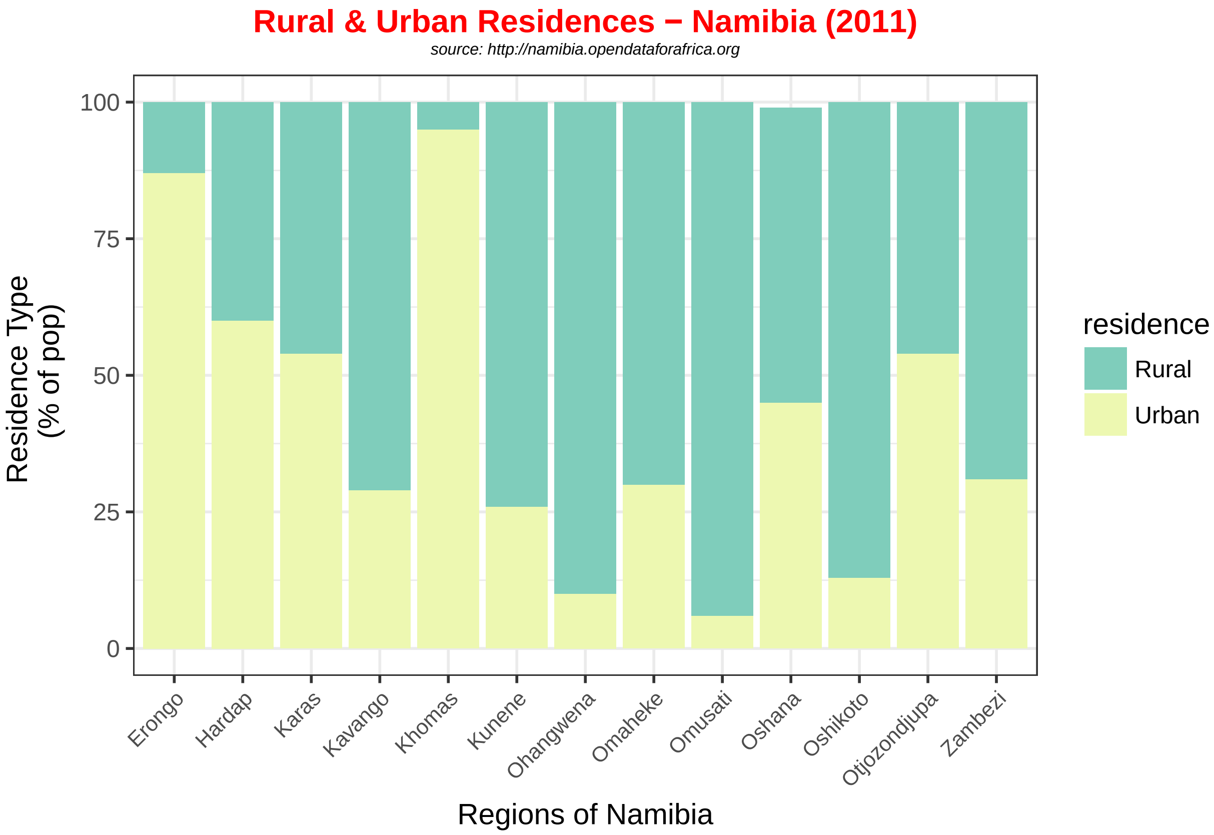
Exercise: I would like you to try improving and changing this plot or one of the earlier ones.

### Changing colour schemes

```
p + scale_fill_brewer(palette = 10)
```



```
p + scale_fill_brewer(palette = 16, direction=-1)
```



## Some additional resources to explore:

- Tutorials from Harvard University (https://dss.iq.harvard.edu/workshop-materials), (http://tutorials.iq.harvard.edu/R/Rgraphics/Rgraphics.html)
- Mapping script available on github entitled: "mappingDataOntoNam_forCourse.R"
- ggplot script exploring some health data from Wales (http://rforbiochemists.blogspot.co.uk/2015/10/exploring-diseases-in-wales-for-sql.html)
- ggplot script to make volcano plots with gene expression data (http://rforbiochemists.blogspot.co.uk/2016/03/gene-expression-analysis-and_7.html)