

Data Science Template

End-to-End **ports** Analysis

Graham Williams

15th September 2018

This template provides an example of a data science template for visualising data. Through visualisation we are able to gain insights into the data and to begin to tell the story that the data supports.

The concept of templates for Data Science are developed in the book [The Essentials of Data Science](#) (2017). The actual source files and scripts, with regular updates, are available from the [Essentials web site](#) (essentials.togaware.com).

As with all our templates and reports we collect up front here the packages used to support the creation of this document.

```
# Load required packages from local library into R.

library(directlabels) # Dodging labels for ggplot2.
library(grid)         # Layout of plots: viewport().
library(magrittr)     # Pipe operator %>% %<>% %T>% equals().
library(rattle)       # normVarNames().
library(readxl)       # read_excel().
library(scales)       # Include commas in numbers.
library(stringi)      # String concat operator %s+%.
library(tidyverse)    # ggplot2, tibble, tidyr, readr, purr, dplyr, stringr
```

1 Data Source

```
# Name of the dataset.

dsname <- "ports"

# Identify the Essentials location of the dataset.

dsloc <- "https://essentials.togaware.com"
dspath <- dsname %s+% ".xlsx"
dsurl <- file.path(dsloc, dspath) %T>% print()

## [1] "https://essentials.togaware.com/ports.xlsx"
```

2 Data Ingestion

```
# Download the file locally.

download.file(dsurl, destfile=dspath, mode="wb")

# Ingest the dataset.

dspath %>%
  read_xlsx(sheet=1, col_names=FALSE) %>%
  assign(dsname, ., envir=.GlobalEnv)

get(dsname)

## # A tibble: 117 x 18
##   X__1 X__2 X__3 X__4 X__5 X__6 X__7 X__8 X__9 X__10 X__11 X__12
##   <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
## 1 1      2      3      4      5      6      7      8      9     10     11     12
## 2 <NA> Adel~ Bris~ Burn~ Damp~ Darw~ Devo~ Frem~ Geel~ Glad~ Hay ~ Melb~
## 3 2011~ 15.5 37     4      176   11     3      28     13     84     83     34
## 4 AvgA~ 4.9   6      3.5   6.5   35     4      4.6    4      6      4.5    5
## 5 <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
## 6 Mixed Bulk <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
## 7 Melb~ Damp~ <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
## 8 Bris~ Glad~ <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
## 9 Port~ Hay ~ <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
## 10 Devo~ Newc~ <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA> <NA>
## # ... with 107 more rows, and 6 more variables: X__13 <chr>, X__14 <chr>,
## #   X__15 <chr>, X__16 <chr>, X__17 <chr>, X__18 <chr>
```

3 Generic Template Variables

```
# Prepare the dataset for usage with template.

ds <- get(dsname)
```

4 Normalise Variable Names

This is not really required for this dataset as we will not be referring to these variable names but we will do so simply to maintain the template.

```
# Normalise the variable names.

names(ds) %<>% normVarNames() %T>% print()

## [1] "x_1" "x_2" "x_3" "x_4" "x_5" "x_6" "x_7" "x_8" "x_9" "x_10"
## [11] "x_11" "x_12" "x_13" "x_14" "x_15" "x_16" "x_17" "x_18"
```

5 Initial Observations

```
# A glimpse into the dataset.
```

```
glimpse(ds)
## Observations: 117
## Variables: 18
## $ x_1 <chr> "1", NA, "2011-12", "AvgAnnualGrowth10yr", NA, "Mixed", "...
## $ x_2 <chr> "2", "Adelaide", "15.5", "4.9", NA, "Bulk", "Dampier", "G...
## $ x_3 <chr> "3", "Brisbane", "37", "6", NA, NA, NA, NA, NA, NA, NA, N...
## $ x_4 <chr> "4", "Burnie", "4", "3.5", NA, NA, NA, NA, NA, NA, NA, NA...
## $ x_5 <chr> "5", "Dampier", "176", "6.5", NA, NA, NA, NA, NA, NA, NA, ...
## $ x_6 <chr> "6", "Darwin", "11", "35", NA, NA, NA, NA, NA, NA, NA, NA...
## $ x_7 <chr> "7", "Devonport", "3", "4", NA, NA, NA, NA, NA, NA, NA, N...
## $ x_8 <chr> "8", "Fremantle", "28", "4.6", NA, NA, NA, NA, NA, NA, NA...
## $ x_9 <chr> "9", "Geelong", "13", "4", NA, NA, NA, NA, NA, NA, NA, NA...
## $ x_10 <chr> "10", "Gladstone", "84", "6", NA, NA, NA, NA, NA, NA, NA, ...
## $ x_11 <chr> "11", "Hay Point", "83", "4.5", NA, NA, NA, NA, NA, NA, N...
## $ x_12 <chr> "12", "Melbourne", "34", "5", NA, NA, NA, NA, NA, NA, NA, ...
## $ x_13 <chr> "13", "Newcastle", "130", "6.5", NA, NA, NA, NA, NA, NA, ...
## $ x_14 <chr> "14", "Port Kembla", "27", "4", NA, NA, NA, NA, NA, NA, N...
## $ x_15 <chr> "15", "Port Hedland", "246", "12.5", NA, NA, NA, NA, NA, ...
## $ x_16 <chr> "16", "Port Walcott", "82", "11", NA, NA, NA, NA, NA, NA, ...
## $ x_17 <chr> "17", "Sydney", "29", "4.5", NA, NA, NA, NA, NA, NA, NA, ...
## $ x_18 <chr> "18", "Townsville", "12.5", "5", NA, NA, NA, NA, NA, NA, ...
```

We note that the spreadsheet contains multiple small tables on the one sheet. We will be treating each table separately.

6 Bar Chart: Value/Weight of Sea Trade

```
# Confirm the row and column span for the table of interest.
```

```
ds[72:93, 1:4]
## # A tibble: 22 x 4
##   x_1      x_2      x_3      x_4
##   <chr> <chr>    <chr> <chr>
## 1 2001-02 Australia 99484 85235
## 2 <NA>    17 Ports 84597 83834
## 3 2002-03 Australia 93429 94947
## 4 <NA>    17 Ports 80170 93566
## 5 2003-04 Australia 89303 93467
## 6 <NA>    17 Ports 76163 92045
## 7 2004-05 Australia 106341 108923
## 8 <NA>    17 Ports 92091 106860
## 9 2005-06 Australia 130856 122211
## 10 <NA>   17 Ports 112278 118779
## # ... with 12 more rows
```

```
# Wrangle the dataset: Rename columns informatively.
```

```
ds[72:93, 1:4] %>%
  set_names(c("period", "location", "export", "import"))
## # A tibble: 22 x 4
##   period location export import
##   <chr> <chr>    <chr> <chr>
## 1 2001-02 Australia 99484 85235
## 2 <NA>    17 Ports 84597 83834
## 3 2002-03 Australia 93429 94947
## 4 <NA>    17 Ports 80170 93566
## 5 2003-04 Australia 89303 93467
## 6 <NA>    17 Ports 76163 92045
## 7 2004-05 Australia 106341 108923
## 8 <NA>    17 Ports 92091 106860
## 9 2005-06 Australia 130856 122211
## 10 <NA>   17 Ports 112278 118779
## # ... with 12 more rows
```

```

# Wrangle the dataset: Numeric variable conversion.

ds[72:93, 1:4] %>%
  set_names(c("period", "location", "export", "import")) %>%
  mutate(
    export = as.numeric(export),
    import = as.numeric(import)
  )

## # A tibble: 22 x 4
##   period location  export import
##   <chr>   <chr>    <dbl> <dbl>
## 1 2001-02 Australia 99484 85235
## 2 <NA>    17 Ports 84597 83834
## 3 2002-03 Australia 93429 94947
## 4 <NA>    17 Ports 80170 93566
## 5 2003-04 Australia 89303 93467
## 6 <NA>    17 Ports 76163 92045
## 7 2004-05 Australia 106341 108923
## 8 <NA>    17 Ports 92091 106860
## 9 2005-06 Australia 130856 122211
## 10 <NA>   17 Ports 112278 118779
## # ... with 12 more rows

# Generate indicies that will be useful for indexing the data.

seq(1,21,2) %>% rep(2) %>% sort()

## [1] 1 1 3 3 5 5 7 7 9 9 11 11 13 13 15 15 17 17 19 19 21 21

# Confirm this achieves the desired outcome.

ds[72:93, 1:4] %>%
  set_names(c("period", "location", "export", "import")) %>%
  extract2("period") %>%
  extract(seq(1,21,2) %>% rep(2) %>% sort())

## [1] "2001-02" "2001-02" "2002-03" "2002-03" "2003-04" "2003-04" "2004-05"
## [8] "2004-05" "2005-06" "2005-06" "2006-07" "2006-07" "2007-08" "2007-08"
## [15] "2008-09" "2008-09" "2009-10" "2009-10" "2010-11" "2010-11" "2011-12"
## [22] "2011-12"

```

```

# Wrangle the dataset: Repair the period column.

ds[72:93, 1:4] %>%
  set_names(c("period", "location", "export", "import")) %>%
  mutate(
    export = as.numeric(export),
    import = as.numeric(import),
    period = period[seq(1, 21, 2) %>% rep(2) %>% sort()]
  )

## # A tibble: 22 x 4
##   period location  export import
##   <chr>   <chr>    <dbl> <dbl>
## 1 2001-02 Australia  99484  85235
## 2 2001-02 17 Ports   84597  83834
## 3 2002-03 Australia  93429  94947
## 4 2002-03 17 Ports   80170  93566
## 5 2003-04 Australia  89303  93467
## 6 2003-04 17 Ports   76163  92045
## 7 2004-05 Australia 106341 108923
## 8 2004-05 17 Ports   92091 106860
## 9 2005-06 Australia 130856 122211
## 10 2005-06 17 Ports  112278 118779
## # ... with 12 more rows

```

```

# Wrangle the dataset: Reshape the dataset.

ds[72:93, 1:4] %>%
  set_names(c("period", "location", "export", "import")) %>%
  mutate(
    export = as.numeric(export),
    import = as.numeric(import),
    period = period[seq(1, 21, 2) %>% rep(2) %>% sort()]
  ) %>%
  gather(type, value, -c(period, location))

## # A tibble: 44 x 4
##   period location  type    value
##   <chr>   <chr>    <chr>  <dbl>
## 1 2001-02 Australia export  99484
## 2 2001-02 17 Ports  export  84597
## 3 2002-03 Australia export  93429
## 4 2002-03 17 Ports  export  80170
## 5 2003-04 Australia export  89303
## 6 2003-04 17 Ports  export  76163
## 7 2004-05 Australia export 106341
## 8 2004-05 17 Ports  export  92091
## 9 2005-06 Australia export 130856
## 10 2005-06 17 Ports  export 112278
## # ... with 34 more rows

```

```

# Identify specific colors required for the organisational style.

cols <- c('#F6A01A', # Primary Yellow
          '#0065A4', # Primary Blue
          '#455560', # Primary Accent Grey
          '#B2BB1E', # Secondary Green
          '#7581BF', # Secondary Purple
          '#BBB0A3', # Secondary Light Grey
          '#E31B23', # Secondary Red
          '#C1D2E8') # Variant Grey

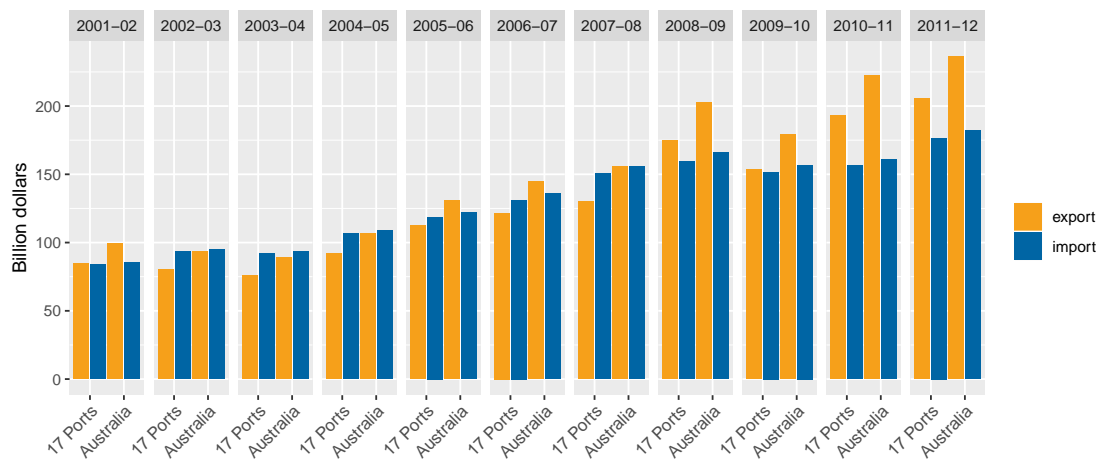
# Create a ggplot2 theme using these colours.

theme_bitre <- scale_fill_manual(values=cols)

# Facetted bar plot comparing import/export value across years.

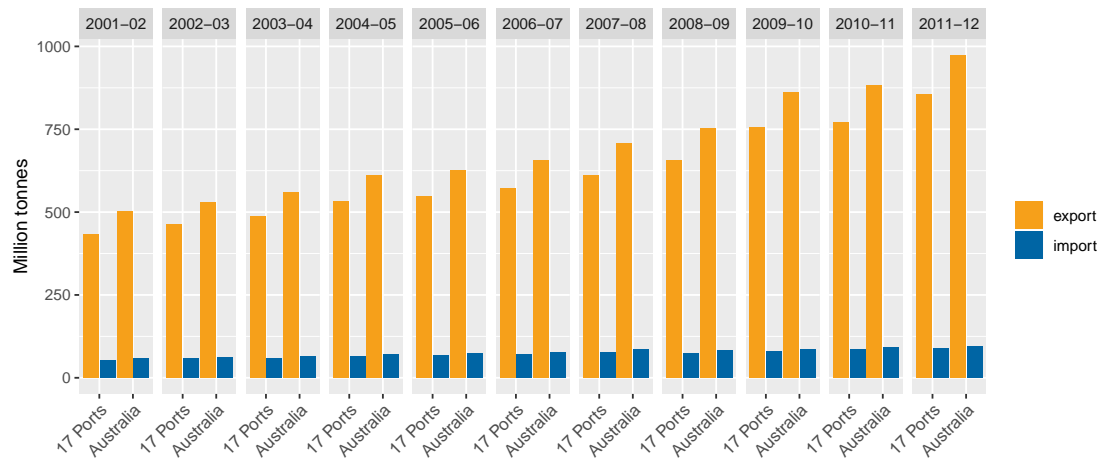
ds[72:93, 1:4] %>%
  set_names(c("period", "location", "export", "import")) %>%
  mutate(
    export = as.numeric(export),
    import = as.numeric(import),
    period = period[seq(1, 21, 2) %>% rep(2) %>% sort()]
  ) %>%
  gather(type, value, -c(period, location)) %>%
  ggplot(aes(x=location, y=value/1000, fill=type)) +
  geom_bar(stat="identity", position=position_dodge(width=1)) +
  facet_grid(~period) +
  labs(y="Billion dollars", x="", fill="") +
  theme(axis.text.x=element_text(angle=45, hjust=1, size=10)) +
  theme_bitre

```



```
# Facetted bar plot comparing import/export weight across years.
```

```
ds[96:117, 1:4] %>%  
  set_names(c("period", "location", "export", "import")) %>%  
  mutate(  
    export = as.numeric(export),  
    import = as.numeric(import),  
    period = period[seq(1, 21, 2) %>% rep(2) %>% sort()]  
  ) %>%  
  gather(type, value, -c(period, location)) %>%  
  ggplot(aes(x=location, y=value/1000, fill=type)) +  
  geom_bar(stat="identity", position=position_dodge(width = 1)) +  
  facet_grid(~period) +  
  labs(y="Million tonnes", x="", fill="") +  
  theme(axis.text.x=element_text(angle=45, hjust=1, size=10)) +  
  theme_bitre
```



7 Scatter Plot: Throughput versus Annual Growth

```
# Confirm the table of interest from the dataset.
```

```
ds[2:4, 2:18]
```

```
## # A tibble: 3 x 17
##   x_2    x_3    x_4    x_5    x_6    x_7    x_8    x_9    x_10    x_11    x_12    x_13
##   <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
## 1 Adela~ Bris~ Burn~ Damp~ Darw~ Devo~ Frem~ Geel~ Glad~ Hay ~ Melb~ Newc~
## 2 15.5   37    4     176   11    3     28   13    84    83    34    130
## 3 4.9    6     3.5   6.5   35    4     4.6   4     6     4.5   5     6.5
## # ... with 5 more variables: x_14 <chr>, x_15 <chr>, x_16 <chr>,
## #   x_17 <chr>, x_18 <chr>
```

```
# Wrangle the dataset: Transpose and retain as a dataset.
```

```
ds[2:4, 2:18] %>%
```

```
  t() %>%
```

```
  data.frame(row.names=NULL, stringsAsFactors=FALSE) %>%
```

```
  tbl_df()
```

```
## # A tibble: 17 x 3
##   X1      X2    X3
##   <chr>   <chr> <chr>
## 1 Adelaide 15.5  4.9
## 2 Brisbane 37     6
## 3 Burnie   4     3.5
## 4 Dampier 176    6.5
## 5 Darwin  11    35
## 6 Devonport 3     4
## 7 Fremantle 28    4.6
## 8 Geelong 13     4
## 9 Gladstone 84     6
## 10 Hay Point 83    4.5
## 11 Melbourne 34     5
## 12 Newcastle 130    6.5
## 13 Port Kembla 27     4
## 14 Port Hedland 246 12.5
## 15 Port Walcott 82    11
## 16 Sydney 29    4.5
## 17 Townsville 12.5  5
```

```

# Wrangle the dataset: Renaming columns informatively.

ds[2:4, 2:18] %>%
  t() %>%
  data.frame(row.names=NULL, stringsAsFactors=FALSE) %>%
  tbl_df() %>%
  set_names(c("port", "weight", "rate"))

## # A tibble: 17 x 3
##   port      weight rate
##   <chr>    <chr> <chr>
## 1 Adelaide  15.5  4.9
## 2 Brisbane  37    6
## 3 Burnie    4     3.5
## 4 Dampier   176   6.5
## 5 Darwin    11    35
## 6 Devonport 3     4
## 7 Fremantle 28    4.6
## 8 Geelong   13    4
## 9 Gladstone 84    6
## 10 Hay Point 83    4.5
## 11 Melbourne 34    5
## 12 Newcastle 130   6.5
## 13 Port Kembla 27    4
## 14 Port Hedland 246  12.5
## 15 Port Walcott 82    11
## 16 Sydney    29    4.5
## 17 Townsville 12.5  5

```

```

# Wrangle the dataset: Numeric variable conversion.

ds[2:4, 2:18] %>%
  t() %>%
  data.frame(row.names=NULL, stringsAsFactors=FALSE) %>%
  tbl_df() %>%
  set_names(c("port", "weight", "rate")) %>%
  mutate(
    weight = as.numeric(weight),
    rate   = as.numeric(rate)
  )

## # A tibble: 17 x 3
##   port      weight rate
##   <chr>      <dbl> <dbl>
## 1 Adelaide    15.5  4.9
## 2 Brisbane     37    6
## 3 Burnie        4   3.5
## 4 Dampier    176   6.5
## 5 Darwin      11   35
## 6 Devonport    3    4
## 7 Fremantle   28   4.6
## 8 Geelong     13    4
## 9 Gladstone   84    6
## 10 Hay Point   83   4.5
## 11 Melbourne   34    5
## 12 Newcastle  130   6.5
## 13 Port Kembla  27    4
## 14 Port Hedland 246  12.5
## 15 Port Walcott  82   11
## 16 Sydney      29   4.5
## 17 Townsville  12.5  5

```

```
# Identify port types from appropriate data in the sheet.
```

```
ds[6, 1:2]
```

```
## # A tibble: 1 x 2  
##   x_1   x_2  
##   <chr> <chr>  
## 1 Mixed Bulk
```

```
ds[7:17, 1:2]
```

```
## # A tibble: 11 x 2  
##   x_1           x_2  
##   <chr>         <chr>  
## 1 Melbourne   Dampier  
## 2 Brisbane    Gladstone  
## 3 Port Kembla Hay Point  
## 4 Devonport   Newcastle  
## 5 Sydney      Port Hedland  
## 6 Geelong     Port Walcott  
## 7 Adelaide    <NA>  
## 8 Fremantle   <NA>  
## 9 Darwin      <NA>  
## 10 Burnie     <NA>  
## 11 Townsville <NA>
```

```
# Construct a port type table.
```

```
ds[7:17, 1:2] %>%
```

```
  set_names(ds[6, 1:2])
```

```
## # A tibble: 11 x 2  
##   Mixed      Bulk  
##   <chr>     <chr>  
## 1 Melbourne Dampier  
## 2 Brisbane  Gladstone  
## 3 Port Kembla Hay Point  
## 4 Devonport Newcastle  
## 5 Sydney    Port Hedland  
## 6 Geelong   Port Walcott  
## 7 Adelaide  <NA>  
## 8 Fremantle <NA>  
## 9 Darwin    <NA>  
## 10 Burnie   <NA>  
## 11 Townsville <NA>
```

```
# Tidy the dataset into a more useful structure.
```

```
ds[7:17, 1:2] %>%  
  set_names(ds[6, 1:2]) %>%  
  gather(type, port) %>%  
  na.omit()
```

```
## # A tibble: 17 x 2  
##   type port  
##   <chr> <chr>  
## 1 Mixed Melbourne  
## 2 Mixed Brisbane  
## 3 Mixed Port Kembla  
## 4 Mixed Devonport  
## 5 Mixed Sydney  
## 6 Mixed Geelong  
## 7 Mixed Adelaide  
## 8 Mixed Fremantle  
## 9 Mixed Darwin  
## 10 Mixed Burnie  
## 11 Mixed Townsville  
## 12 Bulk Dampier  
## 13 Bulk Gladstone  
## 14 Bulk Hay Point  
## 15 Bulk Newcastle  
## 16 Bulk Port Hedland  
## 17 Bulk Port Walcott
```

```

# Wrangle the dataset: Join to port type.

ds[2:4, 2:18] %>%
  t() %>%
  data.frame(row.names=NULL, stringsAsFactors=FALSE) %>%
  tbl_df() %>%
  set_names(c("port", "weight", "rate")) %>%
  mutate(
    weight = as.numeric(weight),
    rate   = as.numeric(rate)
  ) %>%
  left_join(ds[7:17, 1:2] %>%
    set_names(ds[6, 1:2]) %>%
    gather(type, port) %>%
    na.omit(),
    by="port")

## # A tibble: 17 x 4
##   port      weight rate type
##   <chr>      <dbl> <dbl> <chr>
## 1 Adelaide    15.5   4.9 Mixed
## 2 Brisbane     37     6  Mixed
## 3 Burnie       4     3.5 Mixed
## 4 Dampier    176     6.5 Bulk
## 5 Darwin     11    35  Mixed
## 6 Devonport   3     4  Mixed
## 7 Fremantle  28     4.6 Mixed
## 8 Geelong    13     4  Mixed
## 9 Gladstone  84     6  Bulk
## 10 Hay Point  83     4.5 Bulk
## 11 Melbourne  34     5  Mixed
## 12 Newcastle 130     6.5 Bulk
## 13 Port Kembla 27     4  Mixed
## 14 Port Hedland 246    12.5 Bulk
## 15 Port Walcott 82    11  Bulk
## 16 Sydney     29     4.5 Mixed
## 17 Townsville 12.5   5  Mixed

```

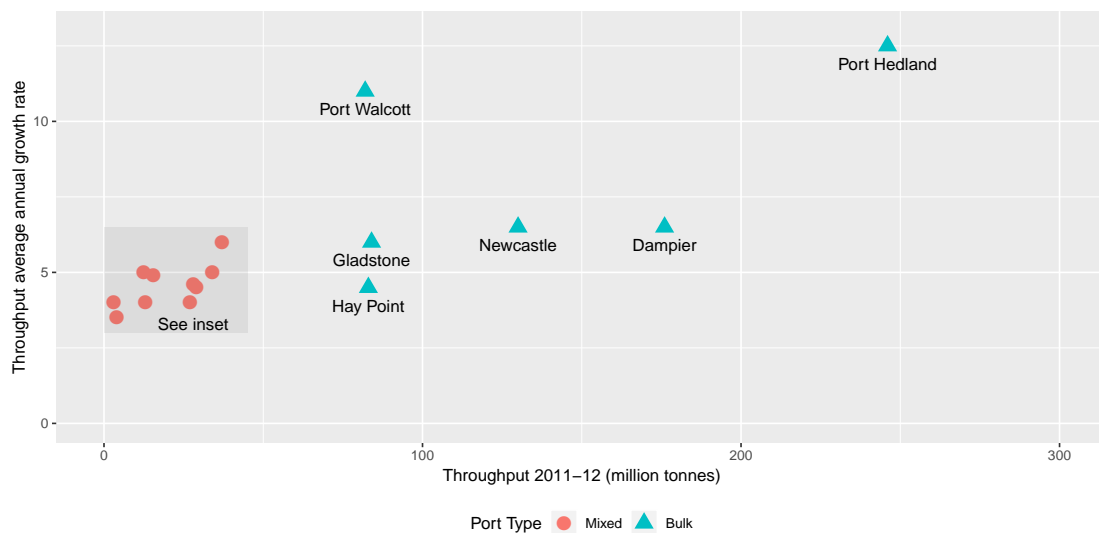
```

# Labelled scatter plot with inset.

ds[2:4, 2:18] %>%
  t() %>%
  data.frame(row.names=NULL, stringsAsFactors=FALSE) %>%
  tbl_df() %>%
  set_names(c("port", "weight", "rate")) %>%
  mutate(weight = as.numeric(weight),
         rate = as.numeric(rate)) %>%
  left_join(ds[7:17, 1:2] %>%
           set_names(ds[6, 1:2]) %>%
           gather(type, port) %>%
           na.omit(),
           by="port") %>%
  mutate(type=factor(type, levels=c("Mixed", "Bulk"))) %>%
  filter(port != "Darwin") ->
tds

tds %>%
  ggplot(aes(x=weight, y=rate)) +
  geom_point(aes(colour=type, shape=type), size=4) +
  xlim(0, 300) + ylim(0, 13) +
  labs(shape="Port Type", colour="Port Type",
       x="Throughput 2011-12 (million tonnes)",
       y="Throughput average annual growth rate") +
  geom_text(data=filter(tds, type=="Bulk"),
           aes(label=port), vjust=2) +
  annotate("rect", xmin=0, xmax=45, ymin=3, ymax=6.5, alpha = .1) +
  annotate("text", label="See inset", x=28, y=3.3, size=4) +
  theme(legend.position="bottom")

```



```

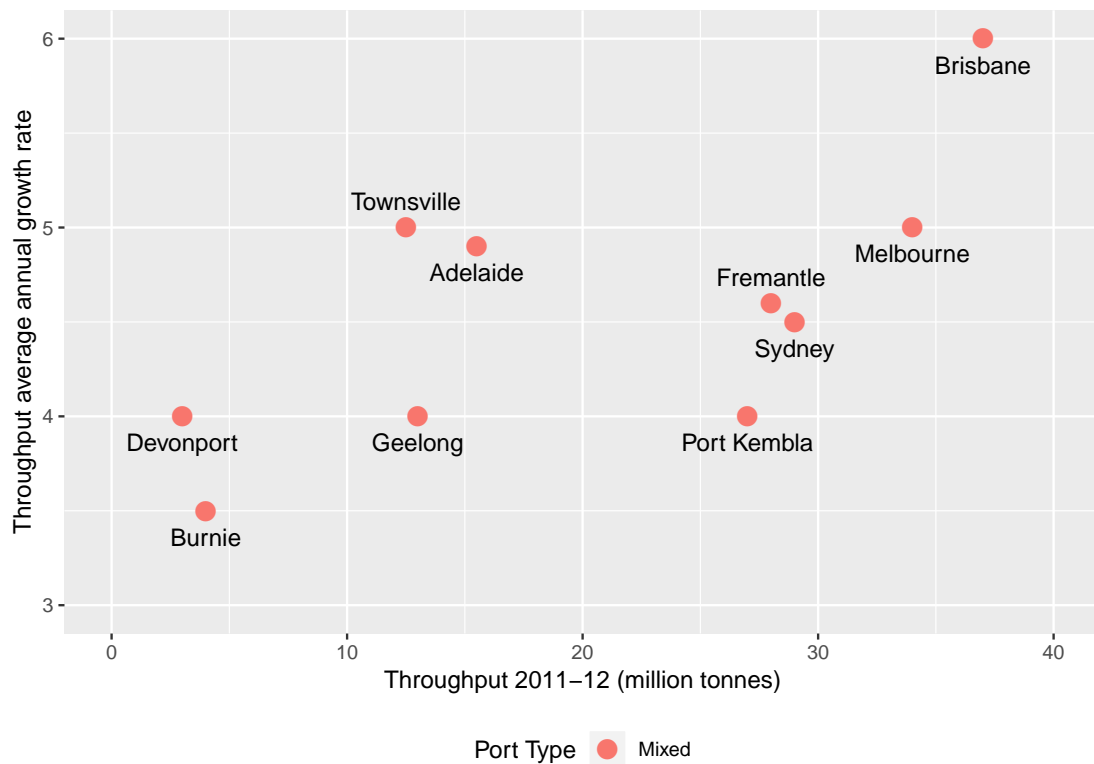
# Labelled scatter plot - the inset.

above <- c("Townsville", "Fremantle")

tds %<>% filter(port != "Darwin" & type == "Mixed")

tds %>%
  ggplot(aes(x=weight, y=rate, label=port)) +
  geom_point(aes(colour=type, shape=type), size=4) +
  labs(shape="Port Type", colour="Port Type") +
  xlim(0, 40) + ylim(3, 6) +
  labs(x="Throughput 2011-12 (million tonnes)",
       y="Throughput average annual growth rate") +
  geom_text(data=filter(tds, !port%in%above), vjust= 2.0) +
  geom_text(data=filter(tds, port%in%above), vjust=-1.0) +
  theme(legend.position="bottom")

```



8 Combined Plots: Port Calls

```
# Wrangle the dataset: Name columns informatively.

ds[20:36, 1:13] %>% set_names(c("port", ds[19, 2:13]))

## # A tibble: 17 x 13
##   port `2001-02` `2002-03` `2003-04` `2004-05` `2005-06` `2006-07`
##   <chr> <chr>    <chr>    <chr>    <chr>    <chr>    <chr>
## 1 Port~ 623      673      547      914      1206     1599
## 2 Melb~ 2628     2902     2935     3061     3088     3159
## 3 Newc~ 1452     1345     1382     1546     1404     1460
## 4 Frem~ 1499     1430     1416     1313     1392     1421
## 5 Glad~ 979      1108     1236     1281     1410     1472
## 6 Damp~ 353      360      698      669      940      1068
## 7 Bris~ 1747     1812     1740     1866     2149     2270
## 8 Sydn~ 1972     1977     2125     2104     2245     2237
## 9 Hay ~ 762      830      944      1044     948      1008
## 10 Adel~ 560      589      649      635      673      657
## 11 Port~ 662      586      609      655      618      641
## 12 Port~ 205      250      304      284      361      386
## 13 Devo~ 581      876      933      966      961      877
## 14 Town~ 541      565      510      455      518      617
## 15 Geel~ 519      479      497      474      446      475
## 16 Darw~ 389      376      305      272      304      344
## 17 Burn~ 470      478      299      464      494      513
## # ... with 6 more variables: `2007-08` <chr>, `2008-09` <chr>,
## #   `2009-10` <chr>, `2010-11` <chr>, `2011-12` <chr>, `2012-13` <chr>

# Wrangle the dataset: Dataset reshape and convert integer.

ds[20:36, 1:13] %>%
  set_names(c("port", ds[19, 2:13])) %>%
  gather(period, calls, -port) %>%
  mutate(calls=as.integer(calls))

## # A tibble: 204 x 3
##   port      period  calls
##   <chr>    <chr>    <int>
## 1 Port Hedland 2001-02   623
## 2 Melbourne   2001-02  2628
## 3 Newcastle   2001-02  1452
## 4 Fremantle   2001-02  1499
## 5 Gladstone   2001-02   979
## 6 Dampier     2001-02   353
## 7 Brisbane    2001-02  1747
## 8 Sydney      2001-02  1972
## 9 Hay Point   2001-02   762
## 10 Adelaide   2001-02   560
## # ... with 194 more rows
```

```

# Wrangle the dataset: Avg calculation.

ds[20:36, 1:13] %>%
  set_names(c("port", ds[19, 2:13])) %>%
  select(port, 2, 13) %>%
  set_names(c('port', 'start', 'end')) %>%
  mutate(
    start = as.integer(start),
    end   = as.integer(end),
    avg   = 100*(exp(log(end/start)/11)-1)
  )

## # A tibble: 17 x 4
##   port      start  end   avg
##   <chr>    <int> <int> <dbl>
## 1 Port Hedland    623  3920  18.2
## 2 Melbourne     2628  3446   2.49
## 3 Newcastle     1452  3273   7.67
## 4 Fremantle     1499  3272   7.35
## 5 Gladstone      979  2857  10.2
## 6 Dampier        353  2855  20.9
## 7 Brisbane     1747  2807   4.41
## 8 Sydney        1972  2536   2.31
## 9 Hay Point      762  1683   7.47
## 10 Adelaide      560  1324   8.14
## 11 Port Kembla    662  1062   4.39
## 12 Port Walcott   205   980  15.3
## 13 Devonport     581   808   3.04
## 14 Townsville    541   648   1.65
## 15 Geelong       519   642   1.95
## 16 Darwin        389   590   3.86
## 17 Burnie        470   410  -1.23

```

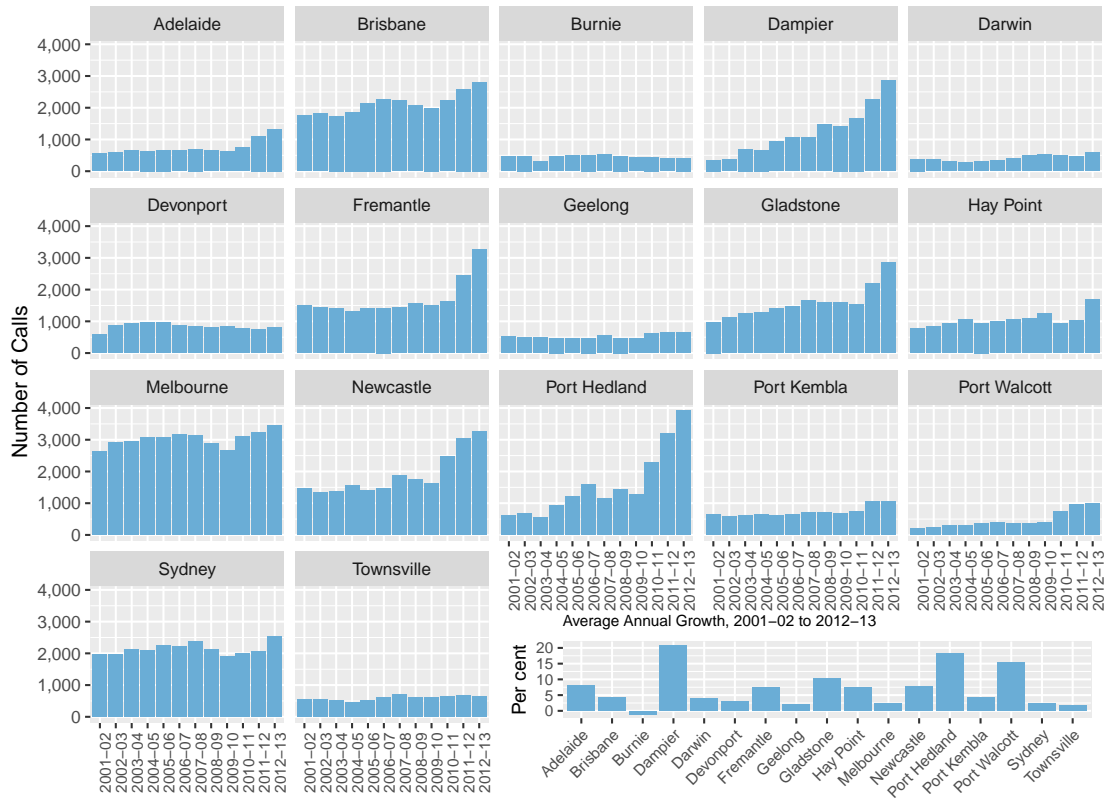
```
# Build the main faceted plot.
```

```
p1 <-  
  ds[20:36, 1:13] %>%  
  set_names(c("port", ds[19, 2:13])) %>%  
  gather(period, calls, -port) %>%  
  mutate(calls=as.integer(calls)) %>%  
  ggplot(aes(x=period, y=calls)) +  
  geom_bar(stat="identity", position="dodge", fill="#6AADD6") +  
  facet_wrap(~port) +  
  labs(x="", y="Number of Calls") +  
  theme(axis.text.x=element_text(angle=90, hjust=1, size=8)) +  
  scale_y_continuous(labels=comma)
```

```
# Generate the second plot.
```

```
p2 <-  
  ds[20:36, 1:13] %>%  
  set_names(c("port", ds[19, 2:13])) %>%  
  select(port, 2, 13) %>%  
  set_names(c('port', 'start', 'end')) %>%  
  mutate(  
    start = as.integer(start),  
    end   = as.integer(end),  
    avg   = 100*(exp(log(end/start)/11)-1)  
  ) %>%  
  ggplot(aes(x=port, y=avg)) +  
  geom_bar(stat="identity",  
          position="identity",  
          fill="#6AADD6") +  
  theme(axis.text.x=element_text(angle=45, hjust=1, size=8),  
        axis.text.y=element_text(size=8),  
        axis.title=element_text(size=10),  
        plot.title=element_text(size=8),  
        plot.background = element_blank()) +  
  labs(x="",  
       y="Per cent",  
       title="Average Annual Growth, 2001-02 to 2012-13")
```

```
# Combine the plots into a single faceted bar plot with embedded bar plot.
print(p1)
print(p2, vp=viewport(x=0.72, y=0.13, height=0.28, width=0.54))
```



9 Horizontal Bar Chart

```
# Horizontal bar chart.
```

```
ds[48:56, 1:2] %>%  
  set_names(c("occupation", "percent")) %>%  
  mutate(percent = as.numeric(percent),  
         occupation = factor(occupation,  
                             levels=occupation[order(percent)])) %>%  
  ggplot(aes(x=occupation, y=percent)) +  
  geom_bar(stat="identity", fill="#6AADD6", width=0.8) +  
  theme(axis.title.x=element_text(size=10)) +  
  labs(x="", y="Per cent") +  
  coord_flip()
```



10 Stacked Horizontal Bar Chart

```
tds <-
  ds[39:40, 2:9] %>%
  set_names(ds[38, 2:9]) %>%
  mutate(type=c("Mixed Ports", "Bulk Ports")) %>%
  gather(occupation, percent, -type) %>%
  mutate(
    percent = as.numeric(percent),
    occupation = factor(occupation,
      levels=c("Managers",
        "Professionals",
        "Technicians and Trades Workers",
        "Community and Personal Service Workers",
        "Clerical and Administrative Workers",
        "Sales Workers",
        "Machinery Operators and Drivers",
        "Labourers"))
  ) %T>%
  print()

## # A tibble: 16 x 3
##   type      occupation      percent
##   <chr>      <fct>          <dbl>
## 1 Mixed Ports Managers          12.2
## 2 Bulk Ports  Managers           8.6
## 3 Mixed Ports Professionals    15.1
## 4 Bulk Ports  Professionals    14.9
## 5 Mixed Ports Technicians and Trades Workers  18.5
## 6 Bulk Ports  Technicians and Trades Workers  28.8
## 7 Mixed Ports Community and Personal Service Workers  5.1
## 8 Bulk Ports  Community and Personal Service Workers  4.4
## 9 Mixed Ports Clerical and Administrative Workers  13.4
## 10 Bulk Ports Clerical and Administrative Workers  12.2
## 11 Mixed Ports Sales Workers           4.3
## 12 Bulk Ports  Sales Workers           2.4
## 13 Mixed Ports Machinery Operators and Drivers  17.6
## 14 Bulk Ports  Machinery Operators and Drivers  14.9
## 15 Mixed Ports Labourers             13.8
## 16 Bulk Ports  Labourers             13.7
```

```

mv <-
  tds %>%
  filter(type=="Mixed Ports") %>%
  extract2("percent") %>%
  rev()

my <- (mv/2) + c(0, head(cumsum(mv), -1))

bv <-
  tds %>%
  filter(type=="Bulk Ports") %>%
  extract2("percent") %>%
  rev()

by <- (bv/2) + c(0, head(cumsum(bv), -1))

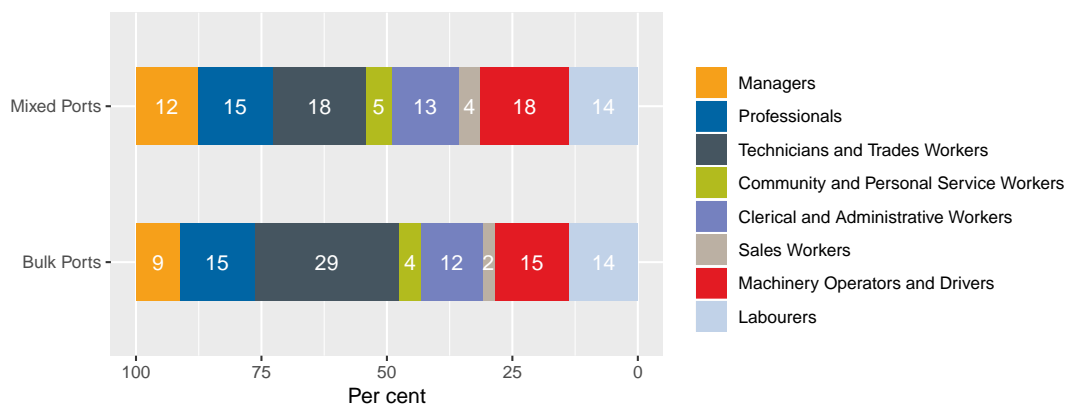
lbls <-
  data.frame(x=c(rep(1, length(mv)), rep(2, length(bv))),
            y=c(by, my),
            v=round(c(bv, mv))) %T>%
  print()

##      x      y  v
## 1  1  6.85 14
## 2  1 21.15 15
## 3  1 29.80  2
## 4  1 37.10 12
## 5  1 45.40  4
## 6  1 62.00 29
## 7  1 83.85 15
## 8  1 95.60  9
## 9  2  6.90 14
## 10 2 22.60 18
## 11 2 33.55  4
## 12 2 42.40 13
## 13 2 51.65  5
## 14 2 63.45 18
## 15 2 80.25 15
## 16 2 93.90 12

```

```
# Horizontal bar chart with multiple stacks.
```

```
tds %>%  
  ggplot(aes(x=type, y=percent, fill=occupation)) +  
  geom_bar(stat="identity", width=0.5) +  
  labs(x="", y="Per cent", fill="") +  
  annotate("text",  
         x=lbls$x,  
         y=lbls$y,  
         label=lbls$v,  
         colour="white") +  
  coord_flip() +  
  scale_y_reverse() +  
  theme_bitre
```




```

# Simple bar chart with dodged and labelled bars.

ds[43:45, 1:3] %>%
  set_names(c("type", ds[42, 2:3])) %>%
  gather(var, count, -type) %>%
  mutate(count = as.integer(count),
         type = factor(type,
                       levels=c("Bulk", "Mixed", "Australia"))) ->
tds

lbls <- data.frame(x=c(.7, 1, 1.3, 1.7, 2, 2.3),
                  y=tds$count-3,
                  lbl=round(tds$count))

tds %>%
  ggplot(aes(x=var, y=count)) +
  geom_bar(stat="identity", position="dodge", aes(fill=type)) +
  labs(x="", y="Per cent", fill="") + ylim(0, 100) +
  geom_text(data=lbls, aes(x=x, y=y, label=lbl), colour="white") +
  theme_bitre

```

