eREAR-mockup / plots / transport-connections-scale-with-volume-styled / **transport-connections.r**

```r
## Map of main freight routes with import/export value in main ports and airports
## Created 20140110 by Y Richard (yvan@dragonfly.co.nz)

library(mbie)
library(rgdal)
library(png)
library(Cairo)
source('../helper-functions.r')
options(warnPartialMatchDollar=F)


## Various parameters for tweaking
xlims <- c(1070000, 2200000)
ylims <- c(4740000, 6200000)
colour_anchor <- design.cols[[2]]
colour_plane  <- design.cols[[1]]
colour_nz          <- grey(0.9)
colour_small_roads <- grey(0.9)
size_small_roads   <- 0.05
size_large_roads   <- 0.7

subplot_size       <- 60000   # width of subplot in map coordinates (map metres)
dollar_scaling     <- 1e9     # scaling imports/exports value from $ to e.g. billion $
volume_scaling     <- 1e6
colour_bar_exp_vol <- design.cols[['cyan']]
colour_bar_imp_vol <- darken(colour_bar_exp_vol, 0.4)
colour_bar_exp_val <- design.cols[['greenish']]
colour_bar_imp_val <- darken(colour_bar_exp_val)
gap_outside_bars   <- 0.1
gap_between_bars   <- 0.08
bars_rel_max_width <- 2.5     # maximum width of the bars, relative to width of logo

legend_x <- 0.15
legend_y <- 0.75
xmin <- 1100000
ymin <- 5700000
minor_y_sep <- 6000
major_y_sep <- 50000
minor_x_sep <- 20000




routes_cols <- c(grey(0.8), lighten(design.cols[6]), darken(design.cols[6], 0.05), design.cols[5])
names(routes_cols) <- c("0 to 1,000", "1,000 to 4,000", "4,000 to 10,000", "More than 10,000")


## Load data
source('../../data/renaming.r')
load('../../data/transport-connections-with-volume/fortified-shapefiles.rdata')
load('../../data/transport-connections-with-volume/imports-exports-year-to-date_impexp.rdata')
ports_loc <- read.csv('ports-locations-2.csv', as.is=T)


theme_blank <- function(base_size = 12, base_family = font_family) {
    theme_bw(base_size = base_size, base_family = base_family) %+replace%
    theme(
        rect = element_blank(),
        line = element_blank(),
        axis.ticks.length = unit(0, "cm"),
        axis.ticks.margin = unit(0, "lines"),
        axis.text = element_blank(),
        axis.ticks = element_blank(),
        axis.title = element_blank(),
        legend.position = 'none'
        )
}

theme_nothing <- function(base_size = 12, base_family = font_family) {
    theme_bw(base_size = base_size, base_family = base_family) %+replace%
    theme(
            panel.background = element_blank(), #element_rect(fill='#00000000'),
```

```
            plot.background = element_blank(), #element_rect(fill='#00000000'),
            panel.grid = element_blank(),
            axis.text = element_blank(),
            axis.ticks = element_blank(),
            axis.title = element_blank(),
            axis.line = element_blank(),
            legend.margin = unit(0, 'mm'),
            panel.margin = unit(0, 'mm'),
            axis.ticks.margin = unit(0, 'mm'),
            axis.ticks.length = unit(0, "mm"),
            legend.margin = unit(0, 'mm'),
            strip.background = element_blank(),
            line = element_blank(),
            rect = element_blank(),
            text = element_blank(),
            legend.key.size = element_blank(),
            title = element_blank(),
            plot.margin = unit.c(unit(0,"line"), unit(0,"line"),
                                 unit(-0.5, "line"),unit(-0.5, "line"))
    )
}

change_logo_col <- function(png, col)
    {
        col <- col2rgb(col) / 255
        png[,,1] <- col[1]
        png[,,2] <- col[2]
        png[,,3] <- col[3]
        return(png)
    }

## Get the two logo bitmaps, change their colour, and "grobise" them
plane_png <- change_logo_col(readPNG('airport.png'), colour_plane)
plane_grob  <- rasterGrob(plane_png)
anchor_png <- change_logo_col(readPNG('port.png'), colour_anchor)
anchor_grob <- rasterGrob(anchor_png)

impexp$import_export <- factor(impexp$import_export, levels=c('export', 'import'))

## Geographical information and offsets
ports_loc <- subset(ports_loc, location %in% impexp$location)
## Convert coordinates from lat/long to NZTM
coordinates(ports_loc) <- ~ x + y
proj4string(ports_loc) <- "+init=epsg:4326"
ports_loc <- as.data.frame(spTransform(ports_loc, CRS("+init=epsg:2193")))
## Apply offset to port locations
ports_loc$x <- ports_loc$x + ports_loc$offset_x
ports_loc$y <- ports_loc$y + ports_loc$offset_y
ports_loc <- ports_loc[, c('location', 'x', 'y', 'name_dir')]

## Split data between trade volume and value
## volume
ie_vol <- subset(impexp, value_volume == 'volume')
ie_vol$value <- ie_vol$value / volume_scaling
ie_vol <- ie_vol[order(ie_vol$location, ie_vol$import_export), ]
## value
ie_val <- subset(impexp, value_volume == 'value')
ie_val$value <- ie_val$value / dollar_scaling
ie_val <- ie_val[order(ie_val$location, ie_val$import_export), ]



## Function to create subplot showing imports/exports next to logo
## returns a grob to be subsequently added to map
make_subplot <- function(loc, max.vol=max(ie_vol$value), max.val=max(ie_val$value))
    {
        if(grepl('Airport', loc))  logo <- plane_grob  else  logo <- anchor_grob

        vol <- subset(ie_vol, location == loc)
        val <- subset(ie_val, location == loc)

        ## Bars for volumes
        bars_vol <- ggplot(vol, aes(x=import_export, y=value, fill=import_export)) +
            geom_bar(stat='identity', width=1-gap_between_bars) +
```

```
            scale_fill_manual(guide='none', values=c(import=colour_bar_imp_vol, export=colour_bar_exp_vol)) +
            coord_flip(ylim=c(0, max.vol)) + # make bars horizontal
            scale_x_discrete(expand=rep(gap_outside_bars, 2)) + # remove gaps on both sides
            scale_y_continuous(expand=c(0,0)) +
            theme_nothing()

        ## Bars for values
        bars_val <- ggplot(val, aes(x=import_export, y=value, fill=import_export)) +
            geom_bar(stat='identity', width=1-gap_between_bars) +
            scale_fill_manual(guide='none', values=c(import=colour_bar_imp_val, export=colour_bar_exp_val)) +
                coord_flip(ylim=c(0, max.val)) + # make bars horizontal
            scale_x_discrete(expand=rep(gap_outside_bars, 2)) + # remove gaps on both sides
            scale_y_reverse(expand=c(0,0)) + # remove gaps on both sides
            theme_nothing()

        ## Construct subplot - mix of grid and ggplot
        subplot <- grid.grabExpr({
            pushViewport(
                viewport(layout = grid.layout(1, 3,
                            widths  = unit(c(bars_rel_max_width, 1, bars_rel_max_width),
                                rep('grobwidth',3), list(plane_grob, plane_grob, plane_grob)),
                            heights = grobHeight(plane_grob))))
            pushViewport(viewport(layout.pos.row = 1, layout.pos.col = 1))
            grid.draw(ggplotGrob(bars_val))
            upViewport()
            pushViewport(viewport(layout.pos.row = 1, layout.pos.col = 2))
            grid.draw(logo)
            upViewport()
            pushViewport(viewport(layout.pos.row = 1, layout.pos.col = 3))
            grid.draw(ggplotGrob(bars_vol))
        })
        return(subplot)
    }


## Create base map (NZ and transport network with traffic)
p <- qplot(xlims, ylims, geom = "blank") +
    geom_polygon(aes(x=long, y=lat, group=id), fill=colour_nz, colour=NA, data=nz) +
    geom_path(aes(x=long, y=lat, group=group), data=hightraff, size=2.5*size_large_roads, colour='black') +
    geom_path(aes(x=long, y=lat, group=group, colour=traff), data=traff, size=size_large_roads) +
    scale_colour_manual(values = routes_cols, guide='none') +
    coord_fixed(ratio=1, xlim=xlims, ylim=ylims) +
    theme_blank()


## Add subplots to base map
subplot_width <- (2*bars_rel_max_width + 1)*subplot_size
p1 <- p
for (l in unique(ie_vol$location)) {
    cat(l, '\n')
    coords <- unique(subset(ports_loc, location==l, select=c('x', 'y')))
    g <- make_subplot(l)
    p1 <- p1 + annotation_custom(g, xmin = coords$x - subplot_width/2, xmax = coords$x + subplot_width/2,
                                ymin = coords$y-subplot_size/2, ymax = coords$y+subplot_size/2)
}


## Name ports
pn.extra <- 5000  ## gap between icon and text
ports_loc$y2 <- ifelse(ports_loc$name_dir=='T', ports_loc$y + subplot_size/2 + pn.extra,
                       ports_loc$y - subplot_size/2 - pn.extra)
ports_loc$x2 <- ports_loc$x
ports_loc$name_vjust <- ifelse(ports_loc$name_dir=='T', 0, 1)
p1.1 <- p1 + gtext(aes(x=x2, y=y2, label=location, vjust=name_vjust), data=ports_loc, hjust=0.5)



###  Add legend manually (tedious, might be simplified?)

leg.line.width <- 0.8*subplot_size
nrcols <- length(routes_cols)
routes_leg <- data.frame(x = rep(c(xmin, xmin+leg.line.width), nrcols),
                         y = rep(ymin + 0:(nrcols-1) * (subplot_size/2 + minor_y_sep), each=2),
                         id = rep(names(routes_cols), each=2))
```

```
## Legend for roads
p1.2 <- p1.1 +
    geom_path(aes(x=x, y=y, group=id), data=routes_leg, colour=rep(routes_cols, each=2), size=1) +
    gtext(aes(x=rep(xmin + leg.line.width + minor_x_sep, nrcols),
                y=ymin + 0:(nrcols-1) * (subplot_size/2 + minor_y_sep),
                label=names(routes_cols)), hjust=0) +
    geom_path(aes(x=c(xmin, xmin+leg.line.width), y=rep(ymin+(nrcols+1)*(subplot_size/2 + minor_y_sep),2)),
                size=2, colour='black') +
    gtext(aes(x=xmin+leg.line.width+minor_x_sep, y=ymin+(nrcols+1)*(subplot_size/2 + minor_y_sep)),
                label='High volume', hjust=0, vjust=0.5) +
    gsmalltext(aes(x=xmin+leg.line.width+minor_x_sep, y=ymin+(nrcols+0.5)*(subplot_size/2 + minor_y_sep),
                label='(over 35,000 vehicles'), hjust=0, vjust=0.9) +
    gsmalltext(aes(x=xmin+leg.line.width+minor_x_sep, y=ymin+(nrcols+0)*(subplot_size/2 + minor_y_sep),
                label='or 1,200 heavy vehicles)'), hjust=0, vjust=0.9) +
    gtext(aes(x=xmin, y=ymin+(nrcols+2)*(subplot_size/2 + minor_y_sep),
                label='STATE HIGHWAYS\n(Vehicles per day)'), hjust=0, vjust=0)


## Legend for logos
ymin2 <- ymin - subplot_size - major_y_sep
p2 <- p1.2 +
    annotation_raster(anchor_png, xmin, xmin+subplot_size, ymin2, ymin2+subplot_size) +
    annotation_raster(plane_png, xmin, xmin+subplot_size, ymin2-subplot_size-minor_y_sep, ymin2-minor_y_sep) +
    gtext(aes(x = xmin + subplot_size + minor_x_sep,
                y = ymin2 + subplot_size/2, label='Seaport'), data=NULL, hjust=0) +
    gtext(aes(x = xmin + subplot_size + minor_x_sep,
                y = ymin2 - minor_y_sep - subplot_size/2, label='Airport'), data=NULL, hjust=0)


###  Legend for import/export value/volume
add_scales <- function() {
    downViewport('panel.3-4-3-4')
    gb <- ggplot_build(p)
    pushViewport(dataViewport(xscale=gb$panel$ranges[[1]]$x.range,
                              yscale=gb$panel$ranges[[1]]$y.range,
                              clip='off'))

    width.mini.bars <- subplot_size/3
    tiny.gap <- 1000

    ##  Trade value
    brks_val <- pretty(c(0, max(ie_val$value)))
    ## title
    y2 <- ymin2 - subplot_size - major_y_sep - minor_y_sep
    grid.text('TRADE VALUE', unit(xmin, 'native'), unit(y2, 'native'), hjust=0, vjust=0.5, gp=gpar_text())
    y3 <- y2 - major_y_sep + minor_y_sep
    ## two bars for import and export
    grid.rect(x = unit(rep(xmin, 2), 'native'),
              y = unit(y3 + c(1, -1) * (width.mini.bars/2 + tiny.gap), 'native'),
              width  = unit(rep(max(brks_val)/max(ie_val$value) * subplot_size*bars_rel_max_width, 2),
                  'native'),
              height = unit(rep(width.mini.bars, 2), 'native'),
              gp     = gpar(col=rep(NA,2), fill=c(colour_bar_imp_val, colour_bar_exp_val)), #
              hjust  = 0, vjust=0.5)
    ## import/export at the end of bars
    grid.text(c('Import', 'Export'),
              x = unit(rep(xmin, 2) + max(brks_val)/max(ie_val$value) * subplot_size*bars_rel_max_width +
                  minor_x_sep, 'native'),
              y = unit(y3 + c(1, -1) * (width.mini.bars/2 + tiny.gap), 'native'),
              gp = gpar_text(), hjust=0, vjust=0.5)
    ## ticks
    grid.polyline(x  = unit(rep(xmin + brks_val/max(ie_val$value) * subplot_size*bars_rel_max_width, each=2),
                      'native'),
                  y  = unit(rep(y3 + c(1, -1) * (tiny.gap + width.mini.bars), length(brks_val)),
                      'native'),
                  id = rep(brks_val, each=2),
                  gp = gpar(col='white', lwd=0.5))
    ## mini axis for scale
    grid.text(brks_val, x=unit(xmin + brks_val/max(ie_val$value) * subplot_size*bars_rel_max_width, 'native'),
              y = unit(y3 - (width.mini.bars + tiny.gap), 'native') - unit(0.5,'mm'),
              gp = gpar_text(), vjust=1)
    grid.text('Billion $',
              x=unit(mean(xmin + brks_val/max(ie_val$value) * subplot_size*bars_rel_max_width), 'native'),
```

```
            y = unit(y3 - (width.mini.bars + tiny.gap), 'native') - unit(3, 'mm'),
            gp = gpar_text(), vjust=1)


    ##  Trade volume
    brks_vol <- pretty(c(0, max(ie_vol$value)))
    y4 <- y3 - major_y_sep - (width.mini.bars + tiny.gap) - 30000
    grid.text('TRADE VOLUME', unit(xmin, 'native'), unit(y4, 'native'), hjust=0, vjust=0.5, gp=gpar_text())
    y5 <- y4 - major_y_sep + minor_y_sep
    ## two bars for import and export
    grid.rect(x = unit(rep(xmin, 2), 'native'),
              y = unit(y5 + c(1, -1) * (width.mini.bars/2 + tiny.gap), 'native'),
              width  = unit(rep(max(brks_vol)/max(ie_vol$value) * subplot_size*bars_rel_max_width, 2),
                  'native'),
              height = unit(rep(width.mini.bars, 2), 'native'),
              gp     = gpar(col=rep(NA,2), fill=c(colour_bar_imp_vol, colour_bar_exp_vol)), #
              hjust  = 0, vjust=0.5)
    ## import/export at the end of bars
    grid.text(c('Import', 'Export'),
              x = unit(rep(xmin, 2) + max(brks_vol)/max(ie_vol$value) * subplot_size*bars_rel_max_width +
                  minor_x_sep,
                  'native'),
              y = unit(y5 + c(1, -1) * (width.mini.bars/2 + tiny.gap), 'native'),
              gp = gpar_text(), hjust=0, vjust=0.5)
    ## ticks
    grid.polyline(x  = unit(rep(xmin + brks_vol/max(ie_vol$value) * subplot_size*bars_rel_max_width, each=2),
                      'native'),
                  y  = unit(rep(y5 + c(1, -1) * (tiny.gap + width.mini.bars), length(brks_vol)),
                      'native'),
                  id = rep(brks_vol, each=2),
                  gp = gpar(col='white', lwd=0.5))
    ## mini axis for scale
    grid.text(brks_vol, x=unit(xmin + brks_vol/max(ie_vol$value) * subplot_size*bars_rel_max_width, 'native'),
              y = unit(y5 - (width.mini.bars + tiny.gap), 'native') - unit(0.5,'mm'),
              gp = gpar_text(), vjust=1)
    grid.text('Million tonnes',
              x=unit(mean(xmin + brks_vol/max(ie_vol$value) * subplot_size*bars_rel_max_width), 'native'),
              y = unit(y5 - (width.mini.bars + tiny.gap), 'native') - unit(3, 'mm'),
              gp = gpar_text(), vjust=1)

    upViewport(0)
}



###  Main plot function
print_plot <- function(p) {
  print(p)
  add_scales()
  ## Add main title
  grid.text('34. Major New Zealand transport network connections 2013', 0.05, 0.98, just='left',
            gp=gpar_title(), vjust=0)
  ## Add sources
  grid.text('Sources: New Zealand Transport Agency', x=0.7, y=unit(0.01,'npc')+unit(1,'lines'),
            hjust=0, vjust=0, gp=gpar_source())
  grid.text('Statistics New Zealand', x=unit(0.7,'npc') + unit(1, 'strwidth', 'Sources: '),
            y=0.01, hjust=0, vjust=0, gp=gpar_source())
}



###  Export plot

W <- 8.3 - 2
H <- 11.7/8.3 * W

CairoPDF('transport-connections.pdf', W, H)
print_plot(p2)
dev.off()

CairoPNG('transport-connections.png', W, H, units='in', res=300)
print_plot(p2)
dev.off()
```