Visualising Data in R

Robin Collis

October 2019

Overview

This Markdown aims to serve as a ggplot cheatsheet for future visualisation tasks. Using the BikeShareData file, the following tasks are to be completed

- Load BikeShareData file, and clean data given there are missing values
- Create the following charts in ggplot to show categories, trends, distributions, and relationships
  - Bar chart
  - Line chart
  - Stacked area
  - Histogram
  - Density plot
  - Boxplot
  - Scatterplot
- Modify each chart to apply the principles of design. Axis labels, titles, subtitles, and captions must be meaningful (placeholder text was used in the example given). Charts must communicate a key insight that is written in the title
- Invent your own chart style. Design aesthetics must be your own. Do not use Avenir font or the fill color (#4cbea3) as shown in the video
- Knit your document to a Word document. The output should include the code and the final visualization for each of the 7 required visuals

Load Packages

```
# Creating a vector of packages used within.
packages <- c('anytime',
              'caTools','chron',
              'DMwR2','doParallel','dplyr',
              'e1071', 'extrafont',
              'ggplot2','gridExtra',
              'here',
              'janitor',
              'knitr',
              'lme4','lubridate',
              'MASS',
              'neuralnet', 'nnet',
              'plyr',
              'randomForest','readr', 'readxl','rlang','rpart','rpart.plot',
              'stats',
              'tidyverse',
              'scales',
              'cowplot',
              'reshape2')
# Checking for package installations on the system and installing if not
found.
if (length(setdiff(packages, rownames(installed.packages()))) > 0) {
  install.packages(setdiff(packages, rownames(installed.packages())))
}
# Including the packages for use.
for(package in packages){
  library(package, character.only = TRUE)
}
#Ensure wd is set to current location by using here()
setwd(here::here())
```

Load and Clean Data

Data is prvided in csv format and is loaded directly from the working directory

The data set has 16 variables across 731 observations. Most of the variablea are numerical in nature, with the date being in character format

```
# Take a look at the data
str(bike_data)

## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 731 obs. of  16
variables:
##  $ instant   : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ dteday    : chr  "01/01/2011" "01/02/2011" "01/03/2011" "01/04/2011"
...
##  $ season    : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ yr        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ mnth      : num  1 1 1 1 1 1 1 1 1 NA ...
##  $ holiday   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ weekday   : num  6 0 1 2 3 4 5 6 0 1 ...
##  $ workingday: num  0 0 1 1 1 1 1 0 0 1 ...
##  $ weathersit: num  2 2 1 1 1 1 2 2 1 1 ...
##  $ temp      : num  0.344 0.363 0.196 0.2 0.227 ...
##  $ atemp     : num  0.364 0.354 0.189 0.212 0.229 ...
##  $ hum       : num  0.806 0.696 0.437 0.59 0.437 ...
##  $ windspeed : num  0.16 0.249 0.248 0.16 0.187 ...
##  $ casual    : num  331 131 120 108 82 88 148 68 54 41 ...
##  $ registered: num  654 670 1229 1454 1518 ...
##  $ cnt       : num  985 801 1349 1562 1600 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   instant = col_double(),
##   ..   dteday = col_character(),
##   ..   season = col_double(),
##   ..   yr = col_double(),
##   ..   mnth = col_double(),
##   ..   holiday = col_double(),
##   ..   weekday = col_double(),
##   ..   workingday = col_double(),
##   ..   weathersit = col_double(),
##   ..   temp = col_double(),
##   ..   atemp = col_double(),
##   ..   hum = col_double(),
##   ..   windspeed = col_double(),
##   ..   casual = col_double(),
##   ..   registered = col_double(),
##   ..   cnt = col_double()
##   .. )
```

```
summary(bike_data)
```

```
##     instant         dteday              season            yr        
##  Min.   :  1.0   Length:731          Min.   :1.000   Min.   :0.0000  
##  1st Qu.:183.5   Class :character   1st Qu.:2.000   1st Qu.:0.0000  
##  Median :366.0   Mode  :character   Median :3.000   Median :1.0000  
##  Mean   :366.0                      Mean   :2.497   Mean   :0.5007  
##  3rd Qu.:548.5                      3rd Qu.:3.000   3rd Qu.:1.0000  
##  Max.   :731.0                      Max.   :4.000   Max.   :1.0000  
##                                                                     
##       mnth           holiday           weekday        workingday   
##  Min.   : 1.000   Min.   :0.00000   Min.   :0.000   Min.   :0.000  
##  1st Qu.: 4.000   1st Qu.:0.00000   1st Qu.:1.000   1st Qu.:0.000  
##  Median : 7.000   Median :0.00000   Median :3.000   Median :1.000  
##  Mean   : 6.527   Mean   :0.02873   Mean   :2.997   Mean   :0.684  
##  3rd Qu.:10.000   3rd Qu.:0.00000   3rd Qu.:5.000   3rd Qu.:1.000  
##  Max.   :12.000   Max.   :1.00000   Max.   :6.000   Max.   :1.000  
##  NA's   :1                                                         
##    weathersit         temp             atemp              hum        
##  Min.   :1.000   Min.   :0.05913   Min.   :0.07907   Min.   :0.0000  
##  1st Qu.:1.000   1st Qu.:0.33708   1st Qu.:0.33784   1st Qu.:0.5200  
##  Median :1.000   Median :0.49833   Median :0.48673   Median :0.6267  
##  Mean   :1.395   Mean   :0.49538   Mean   :0.47435   Mean   :0.6279  
##  3rd Qu.:2.000   3rd Qu.:0.65542   3rd Qu.:0.60860   3rd Qu.:0.7302  
##  Max.   :3.000   Max.   :0.86167   Max.   :0.84090   Max.   :0.9725  
##                                                                     
##    windspeed          casual         registered        cnt     
##  Min.   :0.02239   Min.   :   2.0   Min.   :  20   Min.   :  22  
##  1st Qu.:0.13495   1st Qu.: 315.5   1st Qu.:2497   1st Qu.:3152  
##  Median :0.18097   Median : 713.0   Median :3662   Median :4548  
##  Mean   :0.19049   Mean   : 848.2   Mean   :3656   Mean   :4504  
##  3rd Qu.:0.23321   3rd Qu.:1096.0   3rd Qu.:4776   3rd Qu.:5956
##  Max.   :0.50746   Max.   :3410.0   Max.   :6946   Max.   :8714
##
```

Observign a header view of the data shows that the data structure; for each day over a two year period the number of riders is captured, with a sub-split provided between casual and registered users. Weather metrics are provided each day for temperature, ambient temerature, humidity and windspeed. Metrics related to the date are also provided covering month, yr, season, weekday and workingday

```
# Look at the head of the data
head(bike_data)

## # A tibble: 6 x 16
##   instant dteday season    yr  mnth holiday weekday workingday weathersit
##     <dbl> <chr>   <dbl> <dbl> <dbl>   <dbl>   <dbl>      <dbl>      <dbl>
## 1       1 01/01~      1     0     1       0       6          0          2
## 2       2 01/02~      1     0     1       0       0          0          2
## 3       3 01/03~      1     0     1       0       1          1          1
## 4       4 01/04~      1     0     1       0       2          1          1
## 5       5 01/05~      1     0     1       0       3          1          1
## 6       6 01/06~      1     0     1       0       4          1          1
## # ... with 7 more variables: temp <dbl>, atemp <dbl>, hum <dbl>,
## #   windspeed <dbl>, casual <dbl>, registered <dbl>, cnt <dbl>
```

Checking for missing values show the mnth metric is missing one value

```
# Check number of missing values
sapply(bike_data, function(x) sum(is.na(x)))

##    instant     dteday     season         yr       mnth    holiday
##          0          0          0          0          1          0
##    weekday workingday weathersit       temp      atemp        hum
##          0          0          0          0          0          0
##  windspeed     casual registered        cnt
##          0          0          0          0
```

Correcting the date format of the date columns allows the missing value to be solved for

```
# Convert datacolumn to correct format with lubridate
bike_data$dteday <- mdy(bike_data$dteday)

# Fill in missing to correct format
bike_data$mnth <- month(bike_data$dteday)

# Check number of missing values
sapply(bike_data, function(x) sum(is.na(x)))

##    instant     dteday     season         yr       mnth    holiday
##          0          0          0          0          0          0
##    weekday workingday weathersit       temp      atemp        hum
##          0          0          0          0          0          0
##  windspeed     casual registered        cnt
##          0          0          0          0
```

An additional column is added indicating just the month_year of each day to be used in future analysis at a rollup level

```
# Add month-year column to the data set

bike_data$year_month <- format(bike_data$dteday, "%b-%y")
bike_data$year_month <- as.factor(bike_data$year_month)
bike_data <- as.data.frame(bike_data)
```

In addition a weekday is added in 'weekday' format

```
bike_data$weekday <- strftime(bike_data$dteday, "%A")
```

Finaly, levels are defined so that month_year outputs are ordered correctly

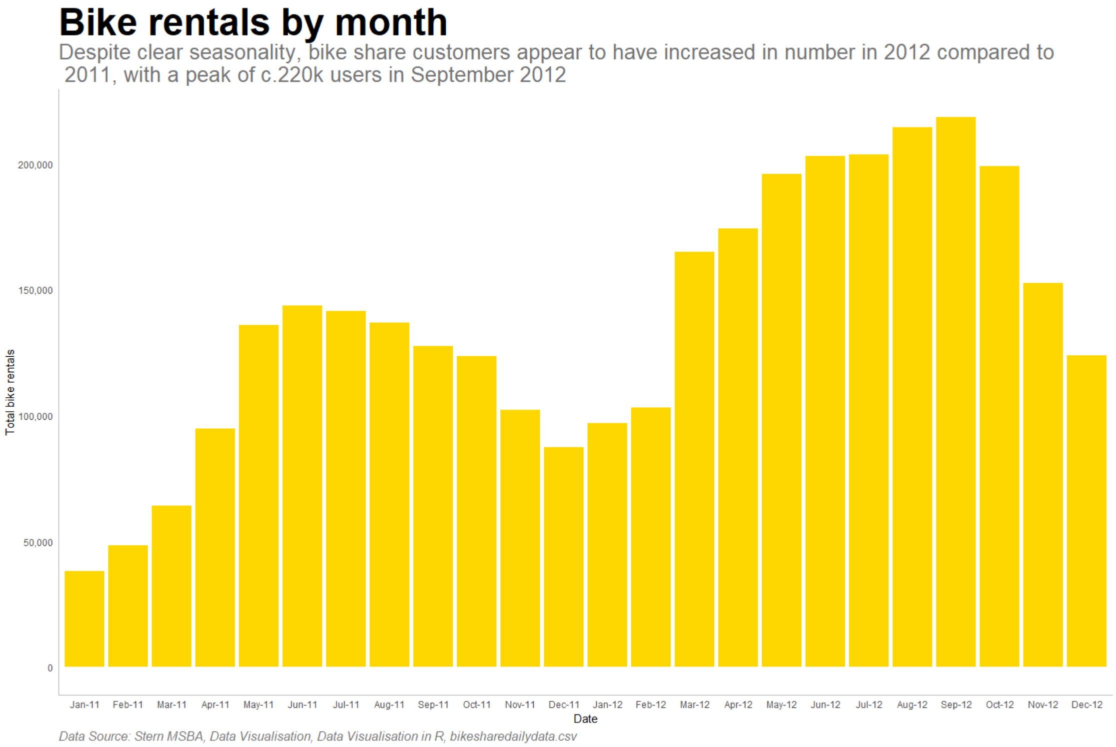
```
# Add levels to maintain order
bike_data$year_month <- factor(bike_data$year_month, levels = c("Jan-11",
"Feb-11", "Mar-11", "Apr-11", "May-11", "Jun-11", "Jul-11", "Aug-11", "Sep-
11", "Oct-11", "Nov-11", "Dec-11",
                                                              "Jan-12", "Feb-
12", "Mar-12", "Apr-12", "May-12", "Jun-12", "Jul-12", "Aug-12", "Sep-12",
"Oct-12", "Nov-12", "Dec-12"))
```

Plot Charts

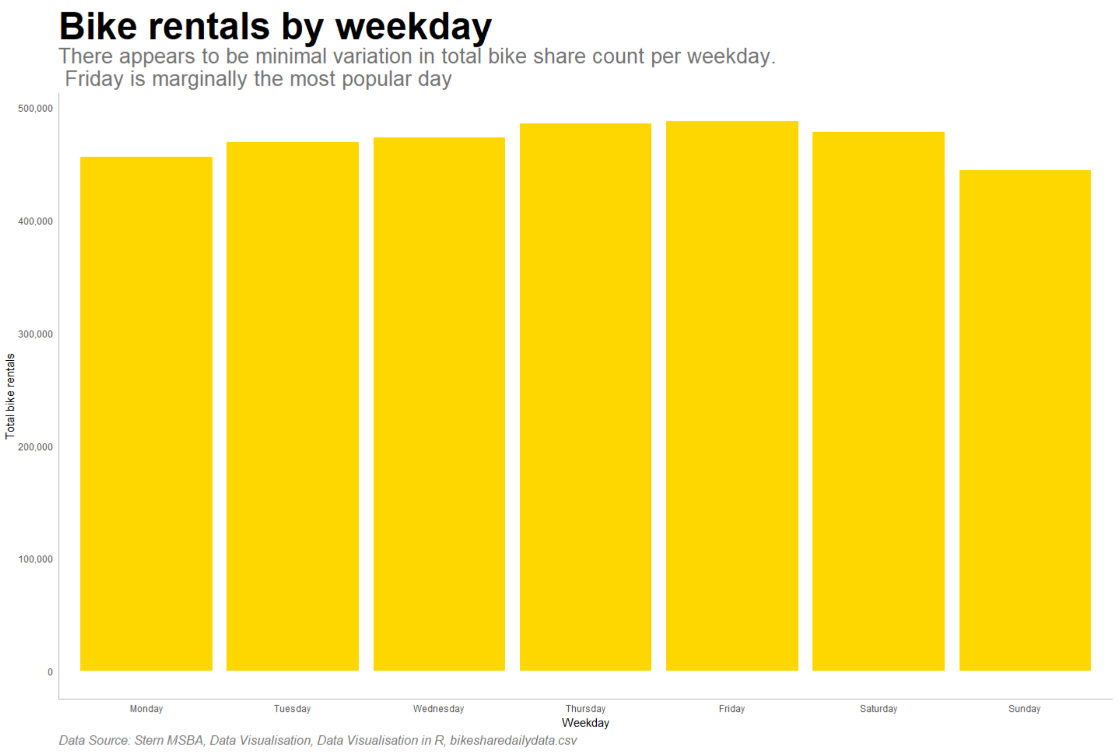
```
# Load windows font calibra
windowsFonts("Calibra" = windowsFont("Calibra"))

# Create RC chart attributes
rc_chartattributes1 <- theme_bw() +
                        theme(text=element_text(family="Calibra")) +
                        theme(panel.border = element_blank(),
                          panel.grid.major = element_blank(),
                          panel.grid.minor = element_blank(),
                          axis.line = element_line(colour = "gray"),
                          axis.ticks.x = element_blank(),
                          axis.ticks.y = element_blank(),
                          plot.title = element_text(color = "black", size =
36, face = "bold"),
                          plot.subtitle = element_text(color = "gray45", size
= 20),
                          plot.caption = element_text(color = "gray45", size
= 12, face = "italic", hjust = 0))
```

Bar Charts

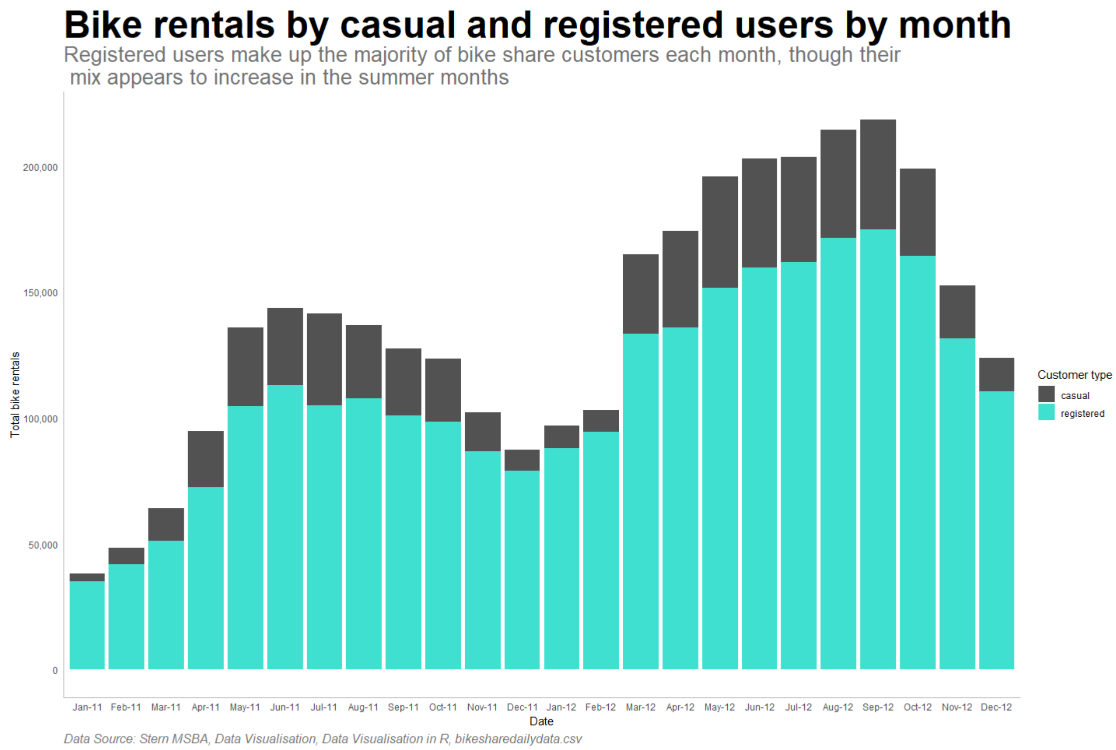
```
# Define bar chart for total bike share users by month
bar_chart_customer_count <- ggplot(data = bike_data) +
                                geom_bar(aes(x = year_month, y = cnt),
stat="identity", fill = "gold1") +
                                labs(title = "Bike rentals by month",
                                      subtitle = "Despite clear seasonality,
bike share customers appear to have increased in number in 2012 compared to
\n 2011, with a peak of c.220k users in September 2012",
                                      caption = "Data Source: Stern MSBA,
Data Visualisation, Data Visualisation in R, bikesharedailydata.csv",
                                      x = "Date",
                                      y = "Total bike rentals") +
                                scale_y_continuous(labels = comma) +
                                rc_chartattributes1

bar_chart_customer_count
```



```
# Set levels
bike_data$weekday <- factor(bike_data$weekday, levels = c("Monday",
"Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"))

# Define bar chart for total bike share users by month
bar_chart_customer_count_weekday <- ggplot(data = bike_data) +
                                geom_bar(aes(x = weekday, y = cnt),
stat="identity", fill = "gold1") +
                                labs(title = "Bike rentals by weekday",
                                      subtitle = "There appears to be minimal
variation in total bike share count per weekday. \n Friday is marginally the
most popular day",
                                      caption = "Data Source: Stern MSBA,
Data Visualisation, Data Visualisation in R, bikesharedailydata.csv",
                                      x = "Weekday",
                                      y = "Total bike rentals") +
                                scale_y_continuous(labels = comma) +
                                rc_chartattributes1

bar_chart_customer_count_weekday
```



```
# Gather data so that segment type forms a customer_type column
bike_data_gather <- gather(bike_data, customer_type, value,
casual:registered)

# Define bar chart for total bike share users by month
bar_chart_customer_type_count <- ggplot(data = bike_data_gather) +
                                  geom_bar(aes(x = year_month, y = value,
fill = customer_type), stat="identity") +
                                  labs(title = "Bike rentals by casual and
registered users by month",
                                      subtitle = "Registered users make up
the majority of bike share customers each month, though their \n mix appears
to increase in the summer months",
                                      caption = "Data Source: Stern MSBA,
Data Visualisation, Data Visualisation in R, bikesharedailydata.csv",
                                      x = "Date",
                                      y = "Total bike rentals",
                                      fill = "Customer type") +
                                  scale_y_continuous(labels = comma) +
                                  scale_fill_manual(values = c("grey32",
"turquoise")) +
                                  rc_chartattributes1

bar_chart_customer_type_count
```



Bike rentals by casual and registered users by month

Registered users make up the majority of bike share customers each month, though their
mix appears to increase in the summer months

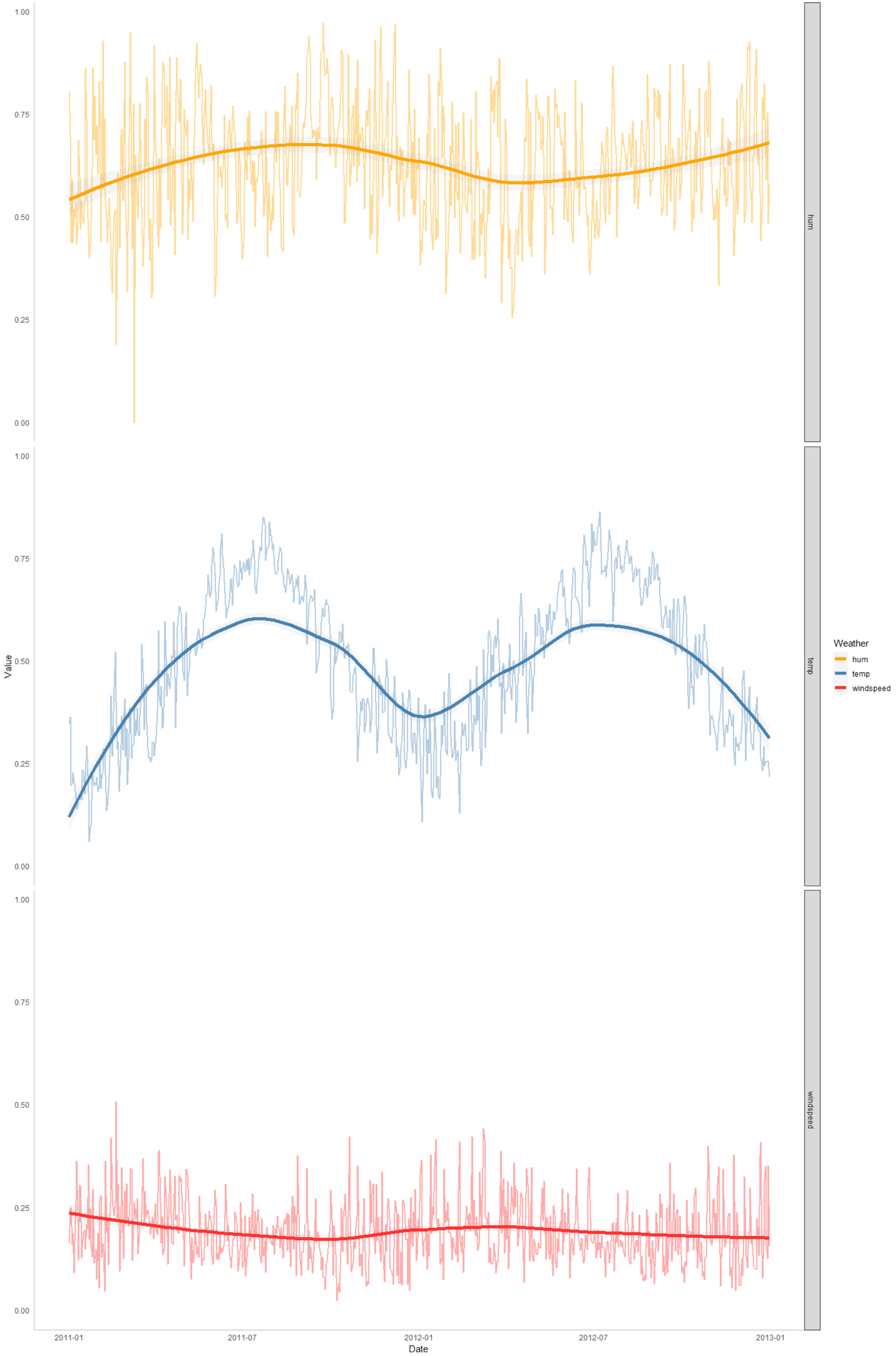
Data Source: Stern MSBA, Data Visualisation, Data Visualisation in R, bikesharedailydata.csv

Line Charts

```
# Gather Data into long form
bike_data_gather_weather <- gather(bike_data, weather, value, temp, hum,
windspeed)

# Facet chart of daily weather across temperature, humidity and windspeed
line_chart_weather <- ggplot(data = bike_data_gather_weather) +
                                geom_line(aes(x = dteday, y = value, color =
weather), alpha = 0.4, size = 1) +
                                geom_smooth(aes(x = dteday, y = value, color
= weather), alpha = 0.1, size = 2) +
                                labs(title = "Weather indicators by day",
                                      subtitle = "Humidity and windspeed
don't appear to fluctuate much over time when smoothed.\n  Temperature
clearly peaks in the summer months",
                                      caption = "Data Source: Stern MSBA,
Data Visualisation, Data Visualisation in R, bikesharedailydata.csv",
                                      x = "Date",
                                      y = "Value",
                                      color = "Weather") +
                                scale_color_manual(values = c("orange",
"steelblue", "firebrick1")) +
                                facet_grid(weather~.) +
                                rc_chartattributes1

line_chart_weather
```

Weather indicators by day

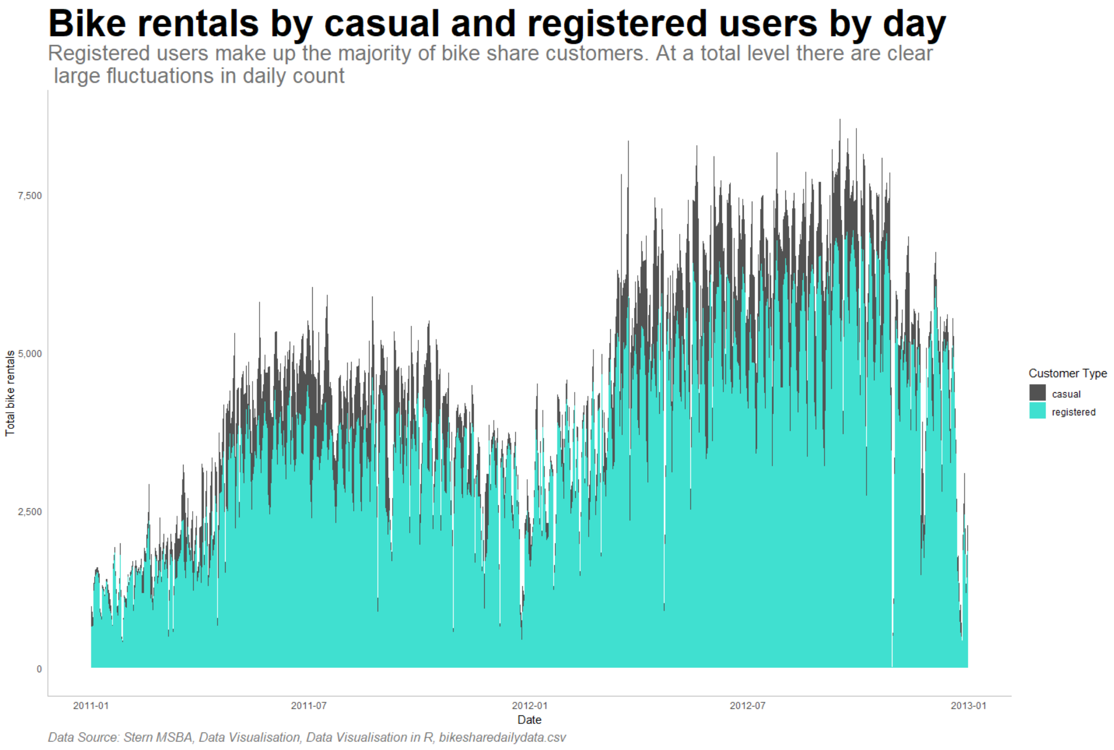
Humidity and windspeed don't appear to fluctuate much over time when smoothed.
Temperature clearly peaks in the summer months



Data Source: Stern MSBA, Data Visualisation, Data Visualisation in R, bikesharedailydata.csv

Stacked Area

```
# Stacked area chart of daily customers by segment
stacked_area_customer_type_count <- ggplot(data = bike_data_gather) +
                                    geom_area(aes(x = dteday, y = value, fill
= customer_type)) +
                                    labs(title = "Bike rentals by casual and
registered users by day",
                                        subtitle = "Registered users make up
the majority of bike share customers. At a total level there are clear \n
large fluctuations in daily count",
                                        caption = "Data Source: Stern MSBA,
Data Visualisation, Data Visualisation in R, bikesharedailydata.csv",
                                        x = "Date",
                                        y = "Total bike rentals",
                                        fill = "Customer Type") +
                                    scale_y_continuous(labels = comma) +
                                    scale_fill_manual(values = c("grey32",
"turquoise")) +
                                    rc_chartattributes1


stacked_area_customer_type_count
```



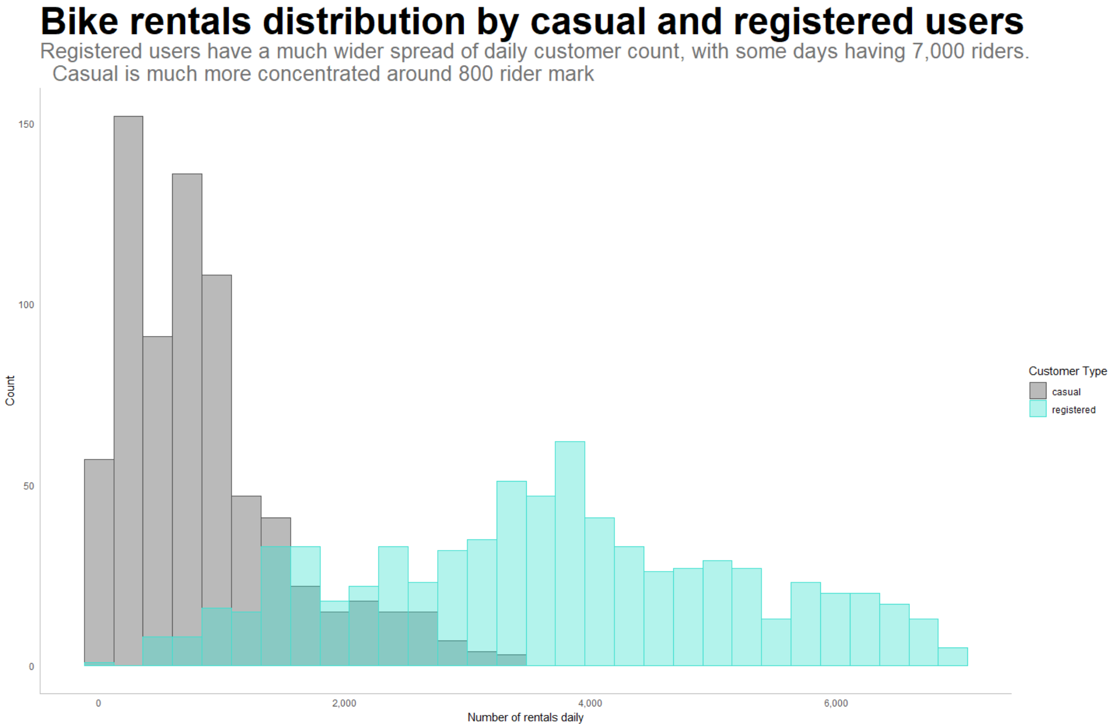
Bike rentals by casual and registered users by day

Registered users make up the majority of bike share customers. At a total level there are clear large fluctuations in daily count

Data Source: Stern MSBA, Data Visualisation, Data Visualisation in R, bikesharedailydata.csv

Histogram

```
histogram_customer_type <- ggplot(data = bike_data_gather) +
                            geom_histogram(aes(x = value, color =
customer_type, fill = customer_type), position = "identity", bins = 30, alpha
= 0.4) +
                            labs(title = "Bike rentals distribution by casual
and registered users",
                                        subtitle = "Registered users have a
much wider spread of daily customer count, with some days having 7,000
riders.\n  Casual is much more concentrated around 800 rider mark",
                                        caption = "Data Source: Stern MSBA,
Data Visualisation, Data Visualisation in R, bikesharedailydata.csv",
                                        x = "Number of rentals daily",
                                        y = "Count",
                                        color = "Customer Type",
                                        fill = "Customer Type") +
                            scale_color_manual(values = c("grey32",
"turquoise")) +
                            scale_fill_manual(values = c("grey32",
"turquoise")) +
                            scale_x_continuous(labels = comma) +
                            rc_chartattributes1

histogram_customer_type
```



Bike rentals distribution by casual and registered users

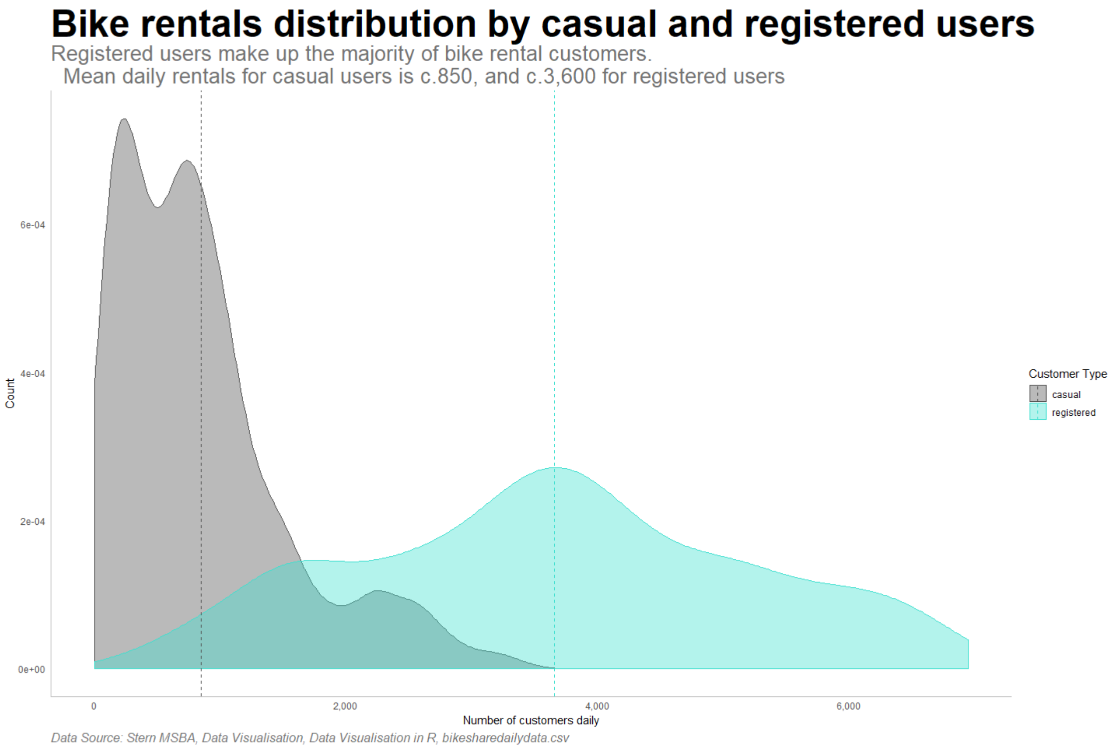
Registered users have a much wider spread of daily customer count, with some days having 7,000 riders.
Casual is much more concentrated around 800 rider mark

Data Source: Stern MSBA, Data Visualisation, Data Visualisation in R, bikesharedailydata.csv

Density Plot

```
# Calculate mean of each group
mean <- ddply(bike_data_gather, "customer_type", summarise,
grp.mean=mean(value))
```

```
# Densirt plot of two customer groups with mean line for each segment
density_plot_customer_type <- ggplot(data = bike_data_gather) +
                            geom_density(aes(x = value, color =
customer_type, fill = customer_type), position = "identity", alpha = 0.4) +
                            labs(title = "Bike rentals distribution by casual
and registered users",
                                        subtitle = "Registered users make up
the majority of bike rental customers.\n  Mean daily rentals for casual users
is c.850, and c.3,600 for registered users ",
                                        caption = "Data Source: Stern MSBA,
Data Visualisation, Data Visualisation in R, bikesharedailydata.csv",
                                        x = "Number of customers daily",
                                        y = "Count",
                                        color = "Customer Type",
                                        fill = "Customer Type") +
                            scale_color_manual(values = c("grey32",
"turquoise")) +
                            scale_fill_manual(values = c("grey32",
"turquoise")) +
                            scale_x_continuous(labels = comma) +
                            geom_vline(data=mean, aes(xintercept = grp.mean,
color=customer_type), linetype="dashed") +
                            rc_chartattributes1

density_plot_customer_type
```

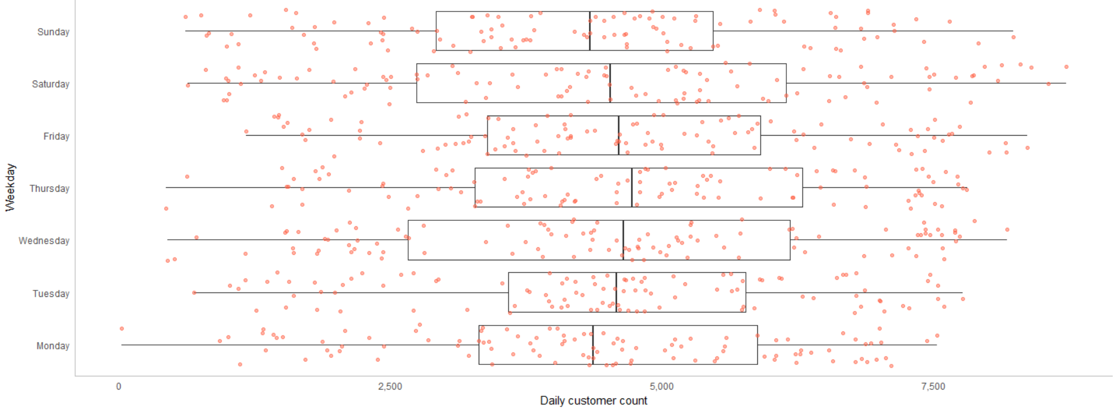


Boxplot

```r
# Set levels
bike_data_gather$weekday <- factor(bike_data_gather$weekday, levels =
c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday",
"Sunday"))


# Total customer count boxplot
boxplot_customer_count <- ggplot(data = bike_data, aes(x = weekday, y = cnt))
+
                            geom_boxplot(alpha = 0) +
                            geom_jitter(alpha = 0.5, color = "tomato") +
                            labs(title = "Bike rentals by weekday: All
riders",
                                        subtitle = "Average rentals is fairly
equal across all weekdays",
                                        caption = "Data Source: Stern MSBA,
Data Visualisation, Data Visualisation in R, bikesharedailydata.csv",
                                        x = "Weekday",
                                        y = "Daily customer count") +
                            scale_y_continuous(labels = comma) +
                            rc_chartattributes1

# Registered customer count boxplot
boxplot_customer_count_registered <- ggplot(data = bike_data, aes(x =
weekday, y = registered)) +
                            geom_boxplot(alpha = 0) +
                            geom_jitter(alpha = 0.5, color = "turquoise") +
                            labs(title = "Bike rentals by weekday: Registered
riders",
                                        subtitle = "Registered customers use
the bike rental service much more during the week, indicating \n the service
as a transport mode for getting to work",
                                        caption = "Data Source: Stern MSBA,
Data Visualisation, Data Visualisation in R, bikesharedailydata.csv",
                                        x = "Weekday",
                                        y = "Daily customer count") +
                            scale_y_continuous(labels = comma) +
                            rc_chartattributes1



# Casual customers count boxplot
boxplot_customer_count_casual <- ggplot(data = bike_data, aes(x = weekday, y
= casual)) +
                            geom_boxplot(alpha = 0) +
                            geom_jitter(alpha = 0.5, color = "grey32") +
                            labs(title = "Bike rentals by weekday: Casual
riders",
                                        subtitle = "Casual customers use the
bike rental service much more at the weekend",
                                        caption = "Data Source: Stern MSBA,
Data Visualisation, Data Visualisation in R, bikesharedailydata.csv",
                                        x = "Weekday",
                                        y = "Daily customer count") +
                            scale_y_continuous(labels = comma) +
                            rc_chartattributes1


plot_grid(boxplot_customer_count + coord_flip(),
boxplot_customer_count_registered + coord_flip(),
boxplot_customer_count_casual + coord_flip(), nrow = 3)
```

Bike rentals by weekday: All riders

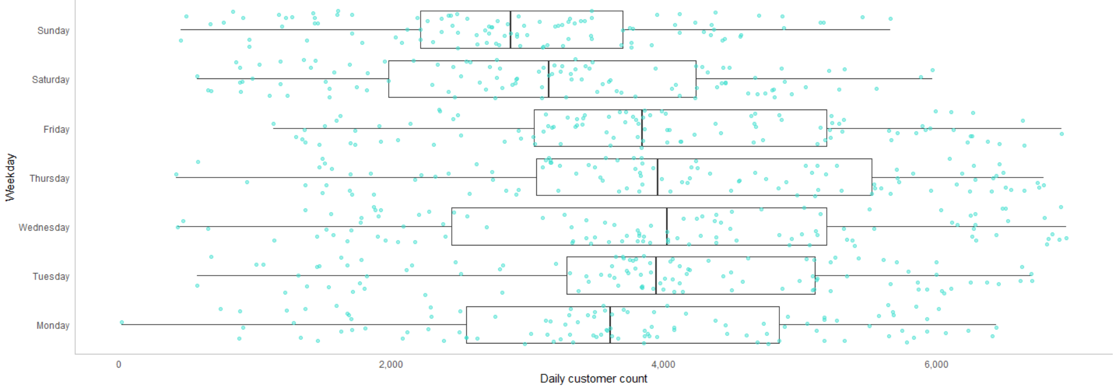
Average rentals is fairly equal across all weekdays



Data Source: Stern MSBA, Data Visualisation, Data Visualisation in R, bikesharedailydata.csv

Bike rentals by weekday: Registered riders

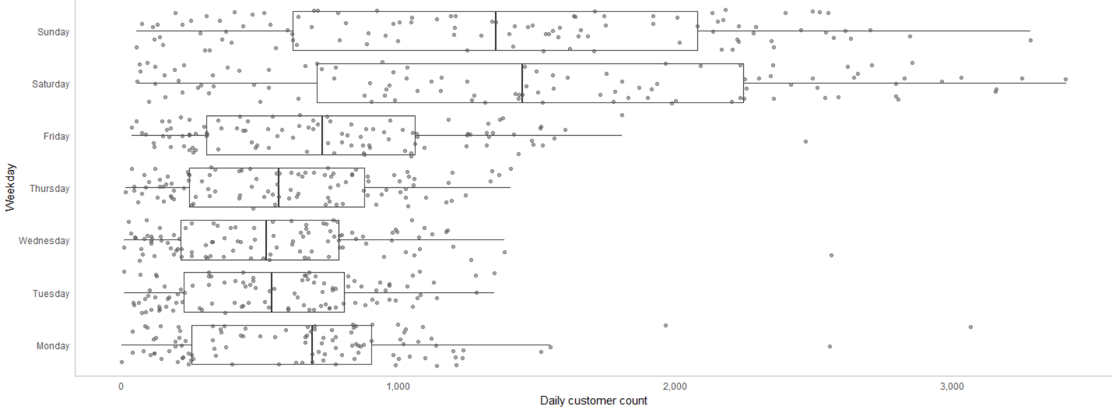
Registered customers use the bike rental service much more during the week, indicating the service as a transport mode for getting to work



Data Source: Stern MSBA, Data Visualisation, Data Visualisation in R, bikesharedailydata.csv

Bike rentals by weekday: Casual riders

Casual customers use the bike rental service much more at the weekend



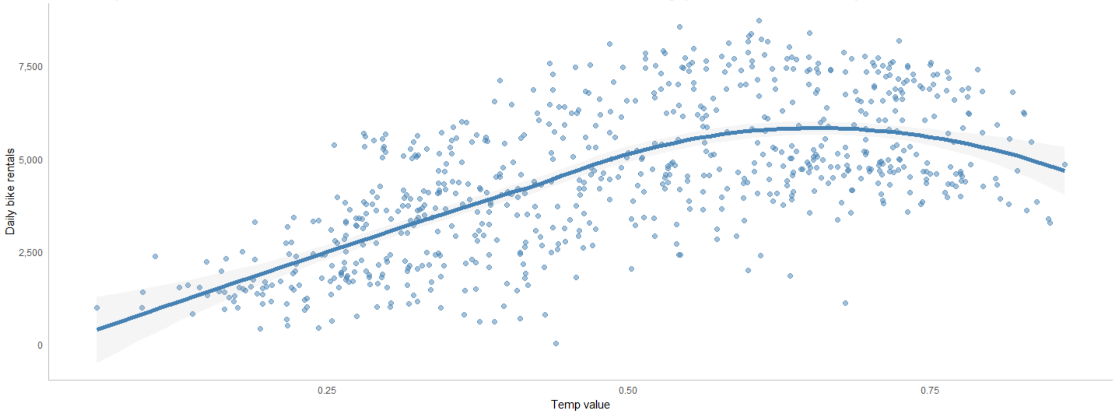
Data Source: Stern MSBA, Data Visualisation, Data Visualisation in R, bikesharedailydata.csv

Scatterplots

```r
# Scatterplot all riders daily count by temperature
scatterplot_temp <- ggplot(data = bike_data) +
                      geom_point(aes(x = temp, y = cnt), color = "steelblue",
size = 2, alpha = 0.5) +
                      geom_smooth(aes(x = temp, y = cnt), color =
"steelblue", alpha = 0.1, size = 2) +
                      labs(title = "Bike rentals and daily temperature",
                                        subtitle = "As temperature increases,
so does number of customers indicating postive relationship",
                                        caption = "Data Source: Stern MSBA,
Data Visualisation, Data Visualisation in R, bikesharedailydata.csv",
                                        x = "Temp value",
                                        y = "Daily bike rentals") +
                      scale_y_continuous(labels = comma) +
                      rc_chartattributes1

# Scatterplot all riders daily count by humidity
scatterplot_hum <- ggplot(data = bike_data) +
                      geom_point(aes(x = hum, y = cnt), color = "orange",
size = 2, alpha = 0.5) +
                      geom_smooth(aes(x = hum, y = cnt), color = "orange",
alpha = 0.1, size = 2) +
                      labs(title = "Bike rentals and daily humidity",
                                        subtitle = "Once humidity reaches a
value of 0.62, the number of daily riders appears to go into decline",
                                        caption = "Data Source: Stern MSBA,
Data Visualisation, Data Visualisation in R, bikesharedailydata.csv",
                                        x = "Humidity value",
                                        y = "Daily bike rentals") +
                      scale_y_continuous(labels = comma) +
                      rc_chartattributes1

# Scatterplot all riders daily count by windspeed
scatterplot_windspeed <- ggplot(data = bike_data) +
                      geom_point(aes(x = windspeed, y = cnt), color =
"firebrick1", size = 2, alpha = 0.5) +
                      geom_smooth(aes(x = windspeed, y = cnt), color =
"firebrick1", alpha = 0.1, size = 2) +
                      labs(title = "Bike rentals and daily windspeed",
                                        subtitle = "Windspeed appears to have
a negative relationship; as speed increases number of daily users declines",
                                        caption = "Data Source: Stern MSBA,
Data Visualisation, Data Visualisation in R, bikesharedailydata.csv",
                                        x = "Windspeed value",
                                        y = "Daily bike rentals") +
                      scale_y_continuous(labels = comma) +
                      rc_chartattributes1

plot_grid(scatterplot_temp, scatterplot_hum, scatterplot_windspeed, nrow = 3)
```

Bike rentals and daily temperature

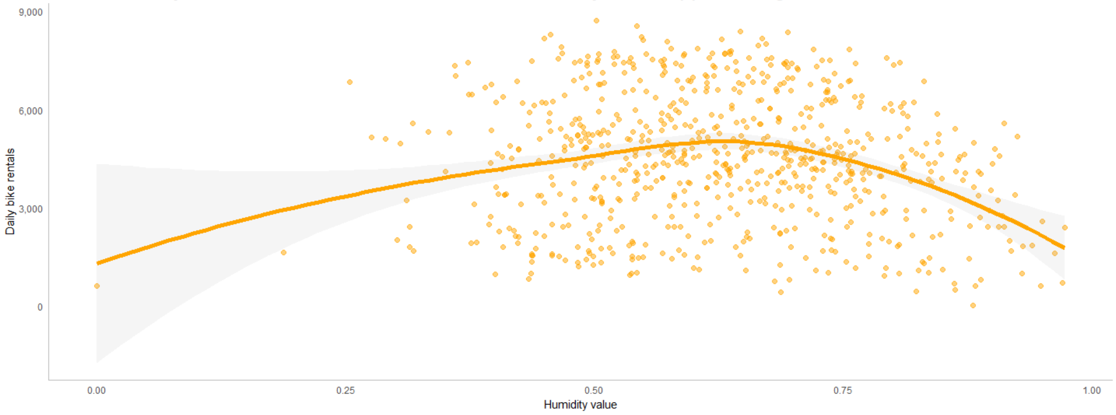
As temperature increases, so does number of customers indicating postive relationship



Data Source: Stern MSBA, Data Visualisation, Data Visualisation in R, bikesharedailydata.csv

Bike rentals and daily humidity

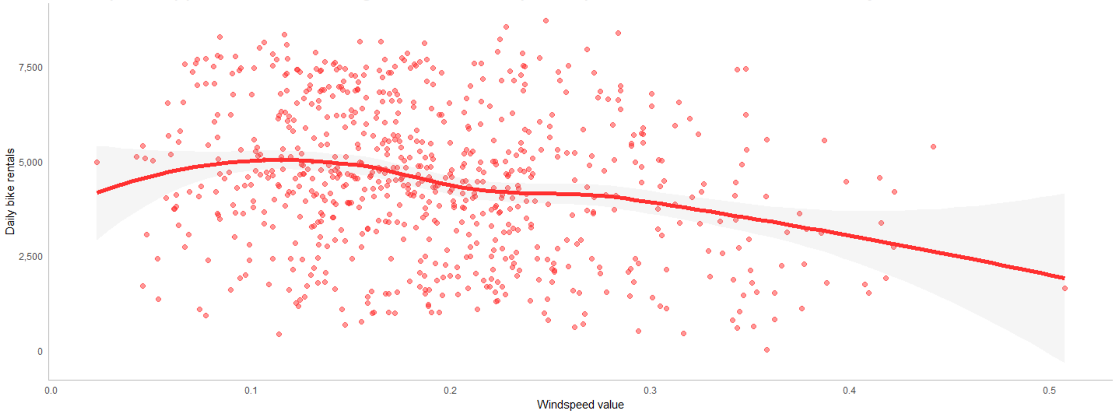
Once humidity reaches a value of 0.62, the number of daily riders appears to go into decline



Data Source: Stern MSBA, Data Visualisation, Data Visualisation in R, bikesharedailydata.csv

Bike rentals and daily windspeed

Windspeed appears to have a negative relationship; as speed increases number of daily users declines



Data Source: Stern MSBA, Data Visualisation, Data Visualisation in R, bikesharedailydata.csv

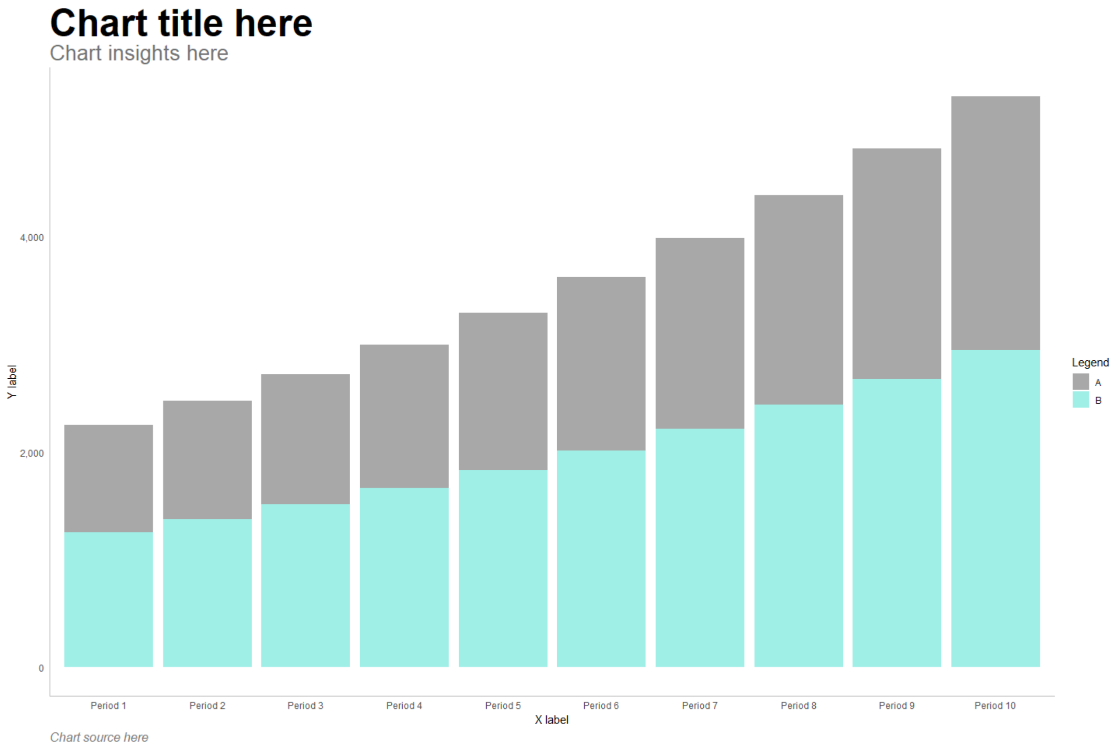
Personal Template

Based on a dummy set of data, I have created a personal template, shown below for a simple stacked bar chart, comprising of the following elements that are part of the function rc_characteristics1

- Fonts set to Calibra
- Chart title set to black, bold and size 22
- Chart subtitle set to gray54 and size 12
- Chart caption, used for displaying the chart source, in grey55 italic, size 10. Position has been adjusted to LHS of the chart
- Bar colours set as grey32 and turquoise, both with alpha of 0.5 to show a level of transparency
- Over ggplot theme of theme_bw utilised
- Panel border, panel grid major and panel grid minor set to element_blank
- axis line set to 'gray'
- Y scale label set to comma

```
# Load template csv file
template_data <- read_csv("template_data.csv", col_names = TRUE)

# View data
template_data

## # A tibble: 20 x 3
##    Period    Class Count
##    <chr>     <chr> <dbl>
##  1 Period 1  A      1000
##  2 Period 1  B      1250
##  3 Period 2  A      1100
##  4 Period 2  B      1375
##  5 Period 3  A      1210
##  6 Period 3  B      1513
##  7 Period 4  A      1331
##  8 Period 4  B      1664
##  9 Period 5  A      1464
## 10 Period 5  B      1830
## 11 Period 6  A      1611
## 12 Period 6  B      2013
## 13 Period 7  A      1772
## 14 Period 7  B      2214
## 15 Period 8  A      1949
## 16 Period 8  B      2436
## 17 Period 9  A      2144
## 18 Period 9  B      2679
## 19 Period 10 A      2358
## 20 Period 10 B      2947
```

```
# Set levels
template_data$Period <- factor(template_data$Period, levels = c("Period 1",
"Period 2", "Period 3", "Period 4", "Period 5", "Period 6", "Period 7",
"Period 8", "Period 9", "Period 10"))

# Define bar chart for total bike share users by month
bar_chart_template <- ggplot(data = template_data) +
                                  geom_bar(aes(x = Period, y = Count, fill =
Class), stat="identity", alpha = 0.5) +
                                  labs(title = "Chart title here",
                                      subtitle = "Chart insights here",
                                      caption = "Chart source here",
                                      x = "X label",
                                      y = "Y label",
                                      fill = "Legend") +
                                  scale_y_continuous(labels = comma) +
                                  scale_fill_manual(values = c("grey32",
"turquoise")) +
                                  rc_chartattributes1

bar_chart_template
```

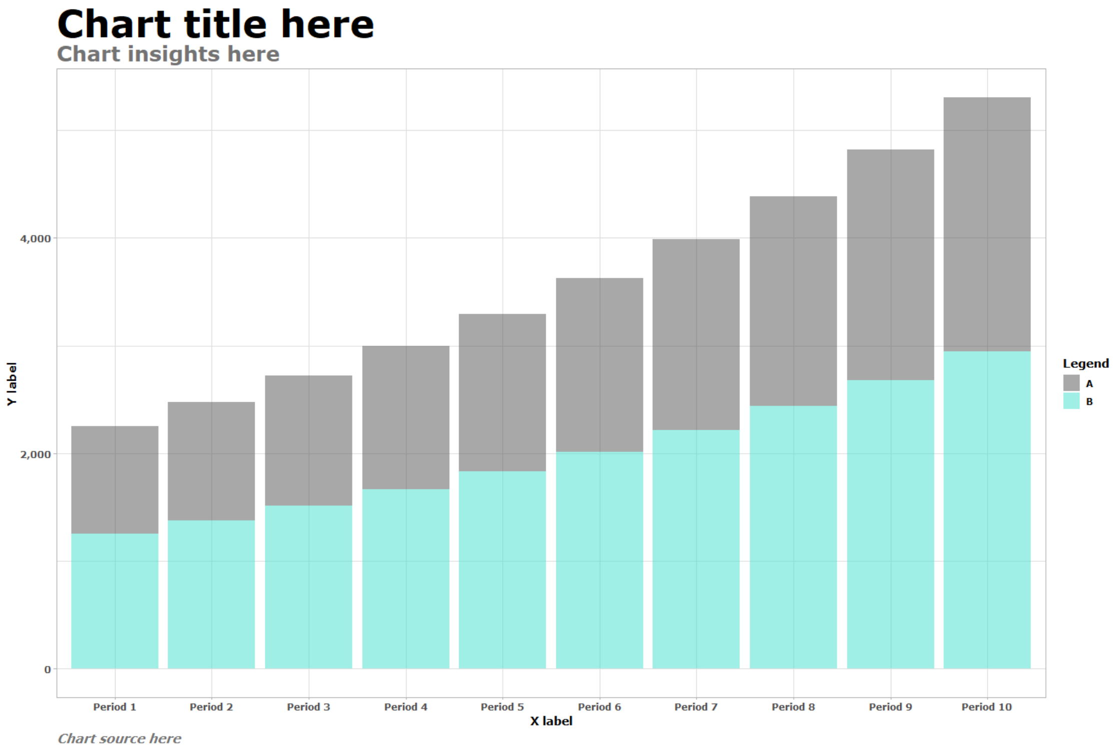


Alternatively, a second personal template using different fonts and maintaining grid lines is:

- Fonts set to Verdana Bold
- Chart title set to black, bold and size 36

- Chart subtitle set to gray54 and size 20
- Chart caption, used for displaying the chart source, in grey55 italic, size 12. Position has been adjusted to LHS of the chart
- Bar colours set as grey32 and turquoise, both with alpha of 0.5 to show a level of transparency
- Over ggplot theme of theme_light
- Y scale label set to comma

```r
# Register fonts for Windows bitmap output
windowsFonts("verdana bold" = windowsFont("verdana bold"))


# Create RC chart attributes
rc_chartattributes2 <- theme_light() +
                        theme(text=element_text(family="verdana bold")) +
                        theme(plot.title = element_text(color = "black", size
= 36, face = "bold"),
                          plot.subtitle = element_text(color = "gray45", size
= 20),
                          plot.caption = element_text(color = "gray45", size
= 12, face = "italic", hjust = 0))


# Define bar chart for total bike share users by month
bar_chart_template2 <- ggplot(data = template_data) +
                                  geom_bar(aes(x = Period, y = Count, fill =
Class), stat="identity", alpha = 0.5) +
                                  labs(title = "Chart title here",
                                      subtitle = "Chart insights here",
                                      caption = "Chart source here",
                                      x = "X label",
                                      y = "Y label",
                                      fill = "Legend") +
                                  scale_y_continuous(labels = comma) +
                                  scale_fill_manual(values = c("grey32",
"turquoise")) +
                                  rc_chartattributes2

bar_chart_template2
```



Alternatively, a third personal template using different fonts and adding background colour is:

- Fonts set to Verdana
- Chart title set to black, bold and size 36
- Chart subtitle set to black and size 20
- Chart caption, used for displaying the chart source, in black italic, size 12. Position has been adjusted to LHS of the chart
- Bar colours set as violetred4 and dodgerblue4, both with alpha of 0.5 to show a level of transparency
- Over ggplot theme of theme_bw
- Panel backgroudn set to fill of #BFD5E3 and colour #6D9EC1 with size and and linetype solid
- Bar colour outline set to white
- Y scale label set to comma

```
# Register fonts for Windows bitmap output
windowsFonts("verdana" = windowsFont("verdana"))


# Create RC chart attributes
rc_chartattributes3 <- theme(text=element_text(family="verdana")) +
                        theme(panel.background = element_rect(fill =
"#BFD5E3", colour = "#6D9EC1", size = 2, linetype = "solid"),
                              panel.grid.major = element_blank(),
                              panel.grid.minor = element_blank(),
                              axis.ticks.x = element_blank(),
                              axis.ticks.y = element_blank(),
                              plot.title = element_text(color = "black", size
= 36, face = "bold", hjust = 0),
                              plot.subtitle = element_text(color = "black",
size = 20),
                              plot.caption = element_text(color = "black",
size = 12, face = "italic", hjust = 0))


# Define bar chart for total bike share users by month
bar_chart_template3 <- ggplot(data = template_data) +
                                  geom_bar(aes(x = Period, y = Count, fill =
Class), colour = "white", stat="identity", alpha = 0.75) +
                                  labs(title = "Chart title here",
                                      subtitle = "Chart insights here",
                                      caption = "Chart source here",
                                      x = "X label",
                                      y = "Y label",
                                      fill = "Legend") +
                                  scale_y_continuous(labels = comma) +
                                  scale_fill_manual(values = c("dodgerblue4",
"violetred4")) +
                                  rc_chartattributes3

bar_chart_template3
```

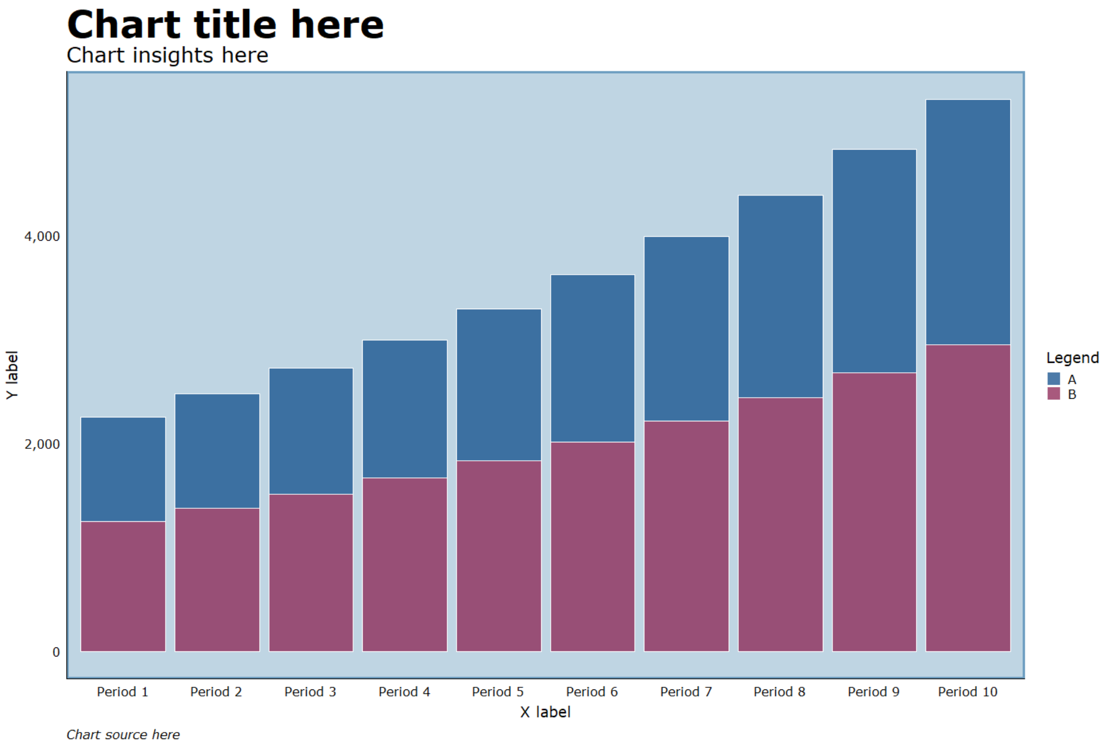


Chart title here
Chart insights here
Y label
Period 1
Period 2
Period 3
Period 4
Period 5
Period 6
Period 7
Period 8
Period 9
Period 10
X label
Legend
A
B
Chart source here