Introduction to ggplot2

Jason Kinyua

1/10/2020

Pre-requisites

- Install required libraries

```r
install.packages(
  c('dplyr','ggplot2','magrittr','quantreg','ggridges'),
  deps=T,
  repos='https://cran.r-project.org'
)
```

- Download datasets
  - Workshop dataset was provided together with other training materials on ICRAFuseR Slack channel
  - The dataset is an extract from Uber Movement
  - Explore more datasets at pathmind

A recap of dplyr grammar

<table>
<thead>
<tr>
<th>Grammar</th>
<th>Description</th>
<th>Use Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>mutate()</td>
<td>adds new variables that are functions of existing variables</td>
<td>mutate(df, col_name=col_1+col_2)</td>
</tr>
<tr>
<td>select()</td>
<td>picks variables based on their names</td>
<td>select(df, -c(&quot;col_1&quot;,&quot;col_2&quot;))</td>
</tr>
<tr>
<td>filter()</td>
<td>picks cases based on their values</td>
<td>filter(df ,col_1&gt;5, col_2 &lt;=20)</td>
</tr>
<tr>
<td>summarise()</td>
<td>reduces multiple values down to a single summary</td>
<td>summarise(df, count=n())</td>
</tr>
<tr>
<td>arrange()</td>
<td>changes the ordering of the rows</td>
<td>arrange(df, desc(date))</td>
</tr>
</tbody>
</table>
| group_by() | allows you to perform any operation “by group” | df %>%
          group_by(column_1) %>%
          summarise(count=n(), mean=mean(column_2, na.rm=TRUE))                                                                                                                                 |

Grammar of Graphics (ggplot) | Components

1. data

- The data used to produce the plot

2. aesthetic mappings
• between variables and visual properties

3. layer(s)

• usually through the geom function to produce geometric shape to be rendered

Geoms for two continuous variables

<table>
<thead>
<tr>
<th>Geoms</th>
<th>Description</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>jitter</td>
<td>Jitter points (to avoid overlapping)</td>
<td>geom_jitter()</td>
</tr>
<tr>
<td>point</td>
<td>Plot points at each x y intersection</td>
<td></td>
</tr>
<tr>
<td>quantile</td>
<td>Plot lines from quantile regression</td>
<td>geom_quantile()</td>
</tr>
<tr>
<td>rug</td>
<td>Plot 1d scatterplot on margins (stripchart)</td>
<td>geom_rug()</td>
</tr>
<tr>
<td>smooth</td>
<td>Plot a smoothing function (many smoothers available)</td>
<td>geom_smooth()</td>
</tr>
<tr>
<td>text</td>
<td>Add text annotations</td>
<td>geom_text()</td>
</tr>
<tr>
<td>bin2d</td>
<td>Bin observations that are close together and color according the density</td>
<td>geom_bin2d()</td>
</tr>
<tr>
<td>density2d</td>
<td>Contour lines of the data density</td>
<td>geom_density2d()</td>
</tr>
<tr>
<td>contours</td>
<td>Surface contour plots</td>
<td>geom_contour()</td>
</tr>
<tr>
<td>hex</td>
<td>Hexagonal bins of data colored according to their density</td>
<td>geom_hex()</td>
</tr>
</tbody>
</table>

Geoms for One variable

<table>
<thead>
<tr>
<th>Geoms</th>
<th>Description</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>area</td>
<td>Filled area plot</td>
<td>geom_area(stat = &quot;bin&quot;)</td>
</tr>
<tr>
<td>density</td>
<td>Density plot</td>
<td>geom_density()</td>
</tr>
<tr>
<td>dotplot</td>
<td>Stacked dotplot, with each dot representing an observation</td>
<td>geom_dotplot()</td>
</tr>
<tr>
<td>polygon of</td>
<td>Polygon of frequencies</td>
<td>geom_freqpoly</td>
</tr>
<tr>
<td>Frequencies</td>
<td></td>
<td></td>
</tr>
<tr>
<td>histogram</td>
<td>Standard histogram</td>
<td>geom_histogram</td>
</tr>
<tr>
<td>barplot</td>
<td>Standard barchart</td>
<td>geom_bar</td>
</tr>
</tbody>
</table>

Enough, Show me the Code!

```r
# Load required libraries
# Uncomment below line to install if not installed
# install.packages(c('dplyr','ggplot2','magrittr'),deps=T,respos='https://cran.r-project.org')
library(dplyr)
library(ggplot2)
library(magrittr)

data <- read.csv('movement-speeds-quarterly-by-hod-nairobi-2019-Q2.csv')
dim(data)
```

## [1] 518005    13
Inspect Dataset Variables

```r
names(data)
```

```r
## [1] "year"     "quarter"   "hour_of_day"
## [4] "segment_id" "start_junction_id" "end_junction_id"
## [7] "osm_way_id" "osm_start_node_id" "osm_end_node_id"
## [10] "speed_kph_mean" "speed_kph_stddev" "speed_kph_p50"
## [13] "speed_kph_p85"
```

Fewer data variables

```r
clean_data <- data %>%
  select(c('hour_of_day', 'osm_start_node_id', 'osm_end_node_id', 'speed_kph_mean'))
```

```r
dim(clean_data)
```

```r
## [1] 518005 4
```

Trim workshop dataset to 500 rows

```r
lab_data <- clean_data %>% slice(c(1:500))
```

```r
dim(lab_data)
```

```r
## [1] 500 4
```

Scatter plot (base `{plot}`)

```r
plot(
  x=lab_data$hour_of_day,
  y=lab_data$speed_kph_mean,
  xlab="Hour", ylab="Speed",
  main="Scatter Plot using base `{plot}`")
```
Scatter Plot using base \{plot\}

```r
ggplot(lab_data, aes(x=hour_of_day, y=speed_kph_mean)) + geom_point() + labs(
  title = "Scatter Plot using ggplot2",
  x = "Hour",
  y = "Speed"
)
```
Scatter Plot using ggplot2

Color Styling ggplot Graphics

```r
ggplot(lab_data, aes(x=hour_of_day, y=speed_kph_mean, color=osm_start_node_id)) +
  geom_point() +
  labs(
    title = "Colored Scatter Plot",
    x = "Hour",
    y = "Speed"
  )
```
Jitter + Smooth Line

```r
ggplot(lab_data, aes(hour_of_day, speed_kph_mean)) +
  geom_jitter() +
  geom_smooth()
```
Travel speed from a given start point

```r
graph <- clean_data %>%
  filter(osm_start_node_id == lab_data$osm_end_node_id[1]) %>%
  ggplot(aes(hour_of_day, speed_kph_mean)) +
  geom_smooth(se=FALSE)
graph
```
Customize labels

```r
labelled <- graph +
xlab("Hour of Day") +
ylab("Travel Mean Speed") +
ggtitle("Travel Speed and Hour of Day")
labelled
```
Travel Speed and Hour of Day

Add Scatter points

pts <- labelled + geom_point(aes(color=osm_end_node_id))
pts
Hex plot

```r
ggplot(lab_data, aes(hour_of_day, speed_kph_mean)) +
  geom_hex() +
  geom_smooth(span=0.2, color='red', size=2, se=FALSE, linetype="dashed")
```
Bin2d Plot

```r
ggplot(lab_data, aes(hour_of_day, speed_kph_mean)) + geom_bin2d()
```
Density2d + Rug Plot

```r
ggplot(lab_data, aes(hour_of_day, speed_kph_mean)) +
  geom_density2d(color='red') +
  geom_rug()
```
10th to 90th quantiles

# Install required package
# install.packages('quantreg',deps=T,respos='https://cran.r-project.org')
ggplot(lab_data,aes(hour_of_day,speed_kph_mean)) +
  geom_quantile(quantiles=seq(0.1,0.9,0.1))
Violin + Boxplot Plots Plot

```r
ggplot(lab_data, aes(hour_of_day, speed_kph_mean)) +
  geom_violin() +
  geom_boxplot() +
  geom_abline(colour = "red", size = 2)
```
Travel Speed Area Graph

```r
speed <- ggplot(lab_data, aes(speed_kph_mean))
speed + geom_area(stat='bin')
```
Frequency Polygon

speed + geom_freqpoly()
Histogram

speed + geom_histogram()
lists + dplyr + ggplot2

geoms <- list(
geom_point(),
geom_abline(),
geom_hex(),
geom_quantile(),
geom_density2d()
)
data %>% filter(osm_start_node_id == lab_data$osm_start_node_id[3]) %>%
    ggplot(aes(hour_of_day, speed_kph_mean, color=hour_of_day)) +
geoms
Off Uber :)

```r
library(datasets)
diamonds %>%
  filter(depth >= 45, depth < 75) %>%
  filter(carat >= 1.5, carat < 4.5) %>%
  ggplot(aes(carat, depth)) +
  geom_line() +
  geom_smooth()
```
# Install required package
#install.packages("ggridges", deps=T, repos="https:// cran. r-project.org")
library(ggridges)
ggplot(mpg, aes(hwy, drv)) +
  geom_density_ridges()
More ridges

```r
ggplot(mpg, aes(hwy, model)) + geom_density_ridges(fill = "cornflowerblue")
```
Colored Bar Plot

```r
ggplot(diamonds, aes(color, fill=cut)) +
  xlab(NULL) + ylab(NULL) +
  theme(legend.position = "none") +
  geom_bar(position = "dodge")
```
Thank you!