Exploratory Data Analysis

Part 1: Tidy data + Univariate graphics

Math 445, Spring 2017

Loading Data into R

Flight Delays

Overview: All departures from LaGuardia during May and June 2009

Variable name	Description
Carrier	UA = United Airlines, AA = American Airlines
FlightNo	Flight number
Destination	Destination airport code
DepartTime	Schedule departure time (4 hr intervals)
Day	Day of the week
Month	May or June
FlightLength	Duration of flight (min.)
Delay	Minutes flight delayed (neg. for early dept.)
Delayed30	Was the flight delayed at least 30 min?

read.table

- If you already have a data set saved, then you can simply load the data set into R.

- Example: If you wanted to read in the `FlightDelays.csv` data set, then run the command (substituting the approriate file path)

  ```
  flights <- read.table(file = "../../data/FlightDelays.csv", sep = ",", header = TRUE)
  ```

- You can use `file.choose()` to get a pop-up window for file selection

  ```
  flights <- read.table(file = file.choose(), sep = ",", header = TRUE)
  ```

  WARNING: This will not work in an R markdown file.

read.table

- `read.table` is our workhorse function, and can read in numerous file types

- for different file types you will need to specify different field separator characters:

Separator	Description
`sep = " "`	white space separated
`sep = "\t"`	tab separated
`sep = ","`	comma separated files (.csv)

- Use `header = TRUE` if there are column names

- `read.csv` is a shortcut to `read.table` where `sep = ","` and `header = TRUE`

Did it work?

The following commands provide useful ways to check that the data loaded correctly

```r
dim(flights)
nrow(flights)
ncol(flights)
str(flights)
head(flights)
```

Textbook data

The `resampledata` R package contains the data sets discussed in the textbook.

```r
# Install (only do once)
install.packages("resampledata")

# Load
library(resampledata)
```

Tidy Data

Data tables

- A row is always a case
- A column is always a variable

```
head(flights)
```

```
##   ID Carrier FlightNo Destination DepartTime Day Month FlightLength Delay Delayed30
## 1  1      UA      403         DEN     4-8am Fri   May          281    -1        No
## 2  2      UA      405         DEN    8-Noon Fri   May          277   102       Yes
## 3  3      UA      409         DEN     4-8pm Fri   May          279     4        No
## 4  4      UA      511         ORD    8-Noon Fri   May          158    -2        No
## 5  5      UA      667         ORD     4-8am Fri   May          143    -3        No
## 6  6      UA      669         ORD     4-8am Fri   May          150     0        No
```

Cases

A case contains all values measured on the same unit across attributes (variables)

```
head(flights)
```

```
##   ID Carrier FlightNo Destination DepartTime Day Month FlightLength Delay Delayed30
## 1  1      UA      403         DEN      4-8am Fri   May          281    -1        No
## 2  2      UA      405         DEN     8-Noon Fri   May          277   102       Yes
## 3  3      UA      409         DEN      4-8pm Fri   May          279     4        No
## 4  4      UA      511         ORD     8-Noon Fri   May          158    -2        No
## 5  5      UA      667         ORD      4-8am Fri   May          143    -3        No
## 6  6      UA      669         ORD      4-8am Fri   May          150     0        No
```

Variables

A variable contains all values that measure the same underlying attribute across cases

- categorical
- quantitative

```
head(flights)
```

```
##   ID Carrier FlightNo Destination DepartTime Day Month FlightLength Delay Delayed30
## 1  1      UA      403         DEN      4-8am Fri   May          281    -1        No
## 2  2      UA      405         DEN     8-Noon Fri   May          277   102       Yes
## 3  3      UA      409         DEN      4-8pm Fri   May          279     4        No
## 4  4      UA      511         ORD     8-Noon Fri   May          158    -2        No
## 5  5      UA      667         ORD      4-8am Fri   May          143    -3        No
## 6  6      UA      669         ORD      4-8am Fri   May          150     0        No
```

Tidy data

1. Each variable forms a column
2. Each case forms a row
3. Each type of case (observational unit) forms a table

```
head(flights)
```

```
##   ID Carrier FlightNo Destination DepartTime Day Month FlightLength Delay Delayed30
## 1  1      UA      403         DEN      4-8am Fri   May          281    -1        No
## 2  2      UA      405         DEN     8-Noon Fri   May          277   102       Yes
## 3  3      UA      409         DEN      4-8pm Fri   May          279     4        No
## 4  4      UA      511         ORD     8-Noon Fri   May          158    -2        No
## 5  5      UA      667         ORD      4-8am Fri   May          143    -3        No
## 6  6      UA      669         ORD      4-8am Fri   May          150     0        No
```

Plotting data

ggplot2

- I prefer using `ggplot2` graphics to the rather `base` graphics system used in the textbook.
- If you are using your personal computer, you will need to install this package before you use it the first time

```r
install.packages("ggplot2")
```

- You will need to load this package at the beginning of each R session:

```r
library(ggplot2)
```

The layered grammar of graphics

- `ggplot2` implements a layered grammar of graphics providing a unified approach to building plots in R

- There is a bit of a learning curve, but the logic behind it is very intuitive

$$\text{base layer} + \text{geometry} + \text{options}$$

- It's easiest to learn by example

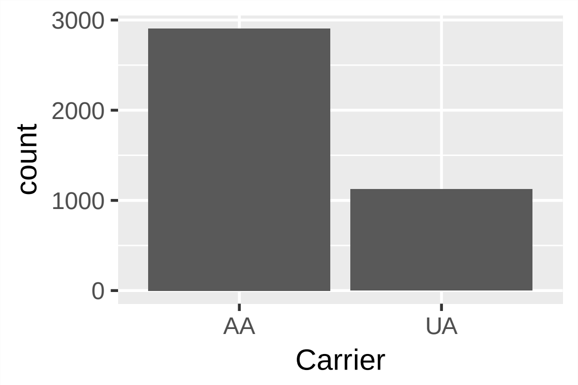
Basic univariate graphics

Variable type	Plot suggestions
Categorical	Bar chart
Quantitative	Histogram
	Boxplot
	Kernel density estimate
	Quantile-quantile plots
	Empirical CDF

Bar charts

Basic bar chart

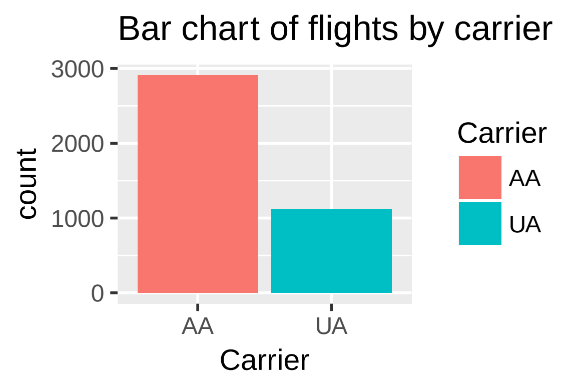
```
ggplot(data = flights, mapping = aes(x = Carrier)) +
  geom_bar()
```



Bar charts

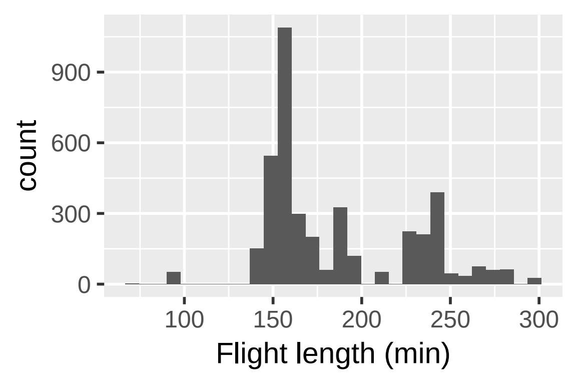
You can also add options

```
ggplot(data = flights, mapping = aes(x = Carrier, fill = Carrier)) +
  geom_bar() +
  labs(title = "Bar chart of flights by carrier")
```



Histograms

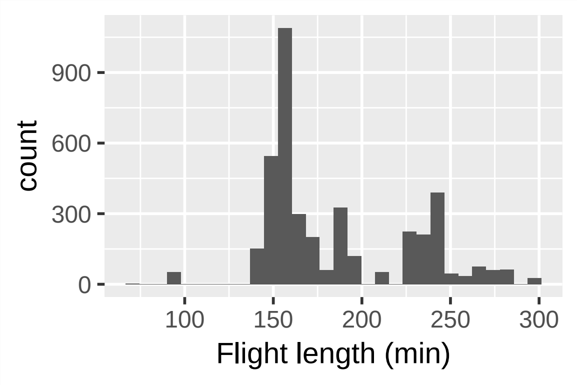
```
ggplot(data = flights, mapping = aes(x = FlightLength)) +
  geom_histogram() +
  labs(x = "Flight length (min)")
```



Always experiment with the bin width

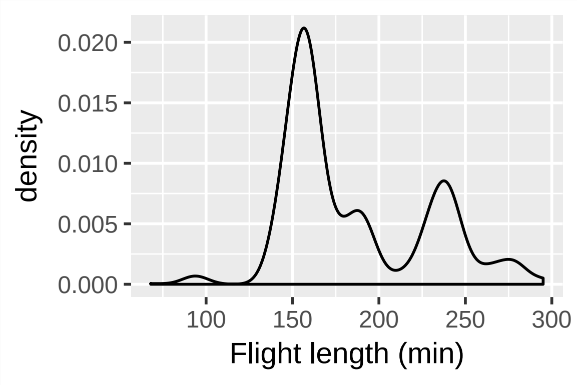
Histograms

```
ggplot(data = flights, mapping = aes(x = FlightLength), binwidth = 30)
  geom_histogram() +
  labs(x = "Flight length (min)")
```



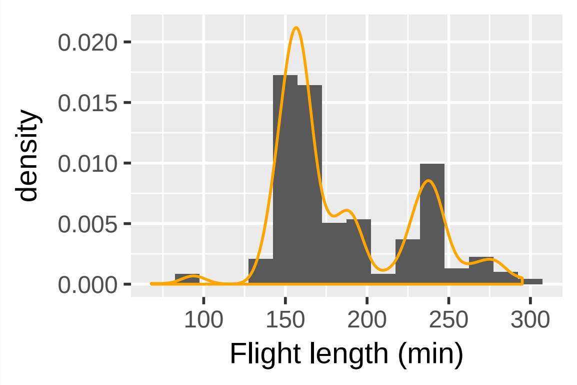
Kernel density estimates

```r
ggplot(data = flights, mapping = aes(x = FlightLength)) +
  geom_density() +
  labs(x = "Flight length (min)")
```



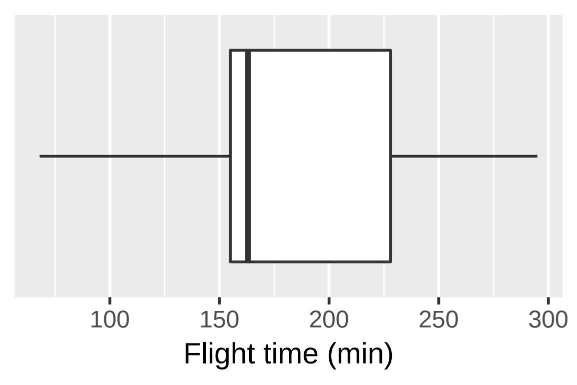
Histograms + Kernel densities

```r
ggplot(data = flights) +
  geom_histogram(mapping = aes(x = FlightLength, y = ..density..), binwidth = 15) +
  geom_density(mapping = aes(x = FlightLength), colour = "orange") +
  labs(x = "Flight length (min)")
```



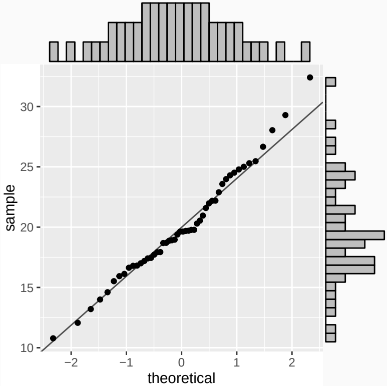
Boxplots

```
ggplot(data = flights, mapping = aes(x = "dummy", y = FlightLength)) +
  geom_boxplot() +
  labs(x = NULL, y = "Flight time (min)") +
  scale_x_discrete(breaks = NULL) +
  coord_flip()
```



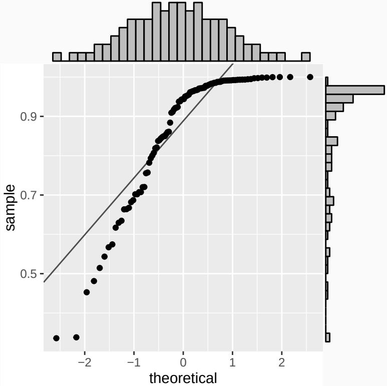
Quantile-quantile plots

- Quantile-quantile (Q-Q) plots compare two sets of quantiles
    - Sample vs. sample
    - Sample vs. theoretical quantiles
- Most common use is for comparison to normality



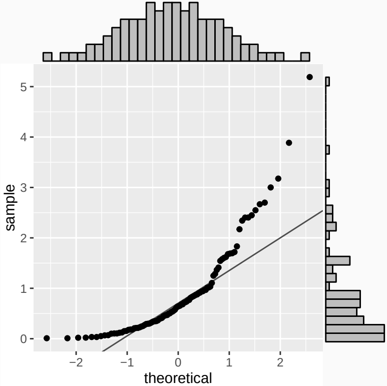
Interpreting Q-Q plots

- Deviations from the diagonal indicate differences between the distributions



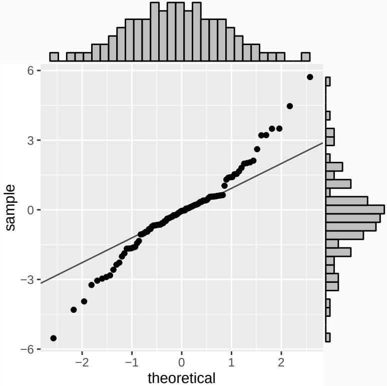
Interpreting Q-Q plots

- Deviations from the diagonal indicate differences between the distributions



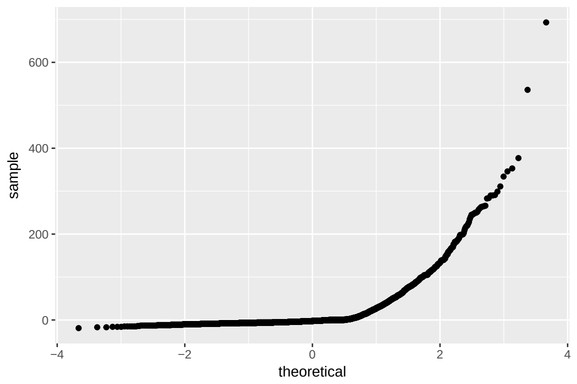
Interpreting Q-Q plots

- Deviations from the diagonal indicate differences between the distributions



Normal Q-Q plots

```r
ggplot(data = flights, mapping = aes(sample = Delay)) +
  geom_point(stat = "qq")
```



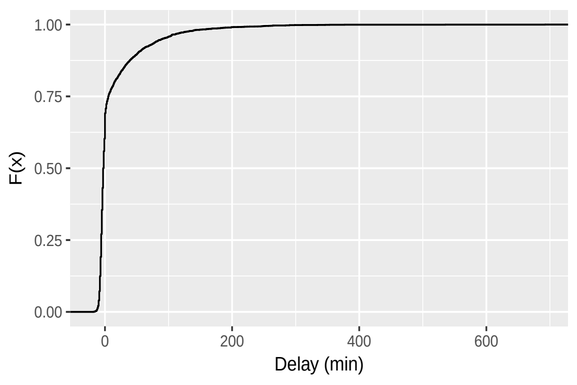
Empirical CDFs

For a sample consisting of n observations $x_1, x_2, \ldots, x_n$, the ECDF is defined as

$$\widehat{F}(x) = \frac{1}{n} \sum_{i=1}^{n} I_{(x_i \leq x)}$$

Empirical CDFs

```r
ggplot(data = flights, mapping = aes(x = Delay)) +
  stat_ecdf(geom = "step") +
  xlab("Delay (min)") +
  ylab("F(x)")
```


F(x)
Delay (min)