DISTRIBUTIONS
Acknowledgements

- Some of these slides have been sourced or modified from slides created by A. Field for Discovering Statistics using R.
GRAPHS
Aims

- How to present data clearly
- Introducing \textit{ggplot2}
- Graphs
  - Scatterplots
  - Histograms
  - Boxplots
  - Error bar charts
  - Line graphs
Aims

- How to present data clearly
- Introducing ggplot2
- Graphs
  - Scatterplots
  - Histograms
  - Boxplots
  - Error bar charts
  - Line graphs
The Art of Presenting Data

- Keep graphs as simple as possible – avoid clutter.
- Use a graph style that matches the type of data you are depicting.
- Ensure that:
  - Lines are thick enough to be easily visible
  - Symbols and fonts are large enough to be easily readable
  - Colours are distinct enough to be easily distinguished
Why Is This Graph Bad?

Error Bars show 95.0 % CI of Mean

Bars show Means
Why Is This Graph Better?

- **Number of Obsessive Thoughts/Actions per Day**
  - Therapy Types: CBT, BT, No Treatment
  - Error Bars Show 95% CI

Many projects will involve the production of several graphs.

You can view these graphs simultaneously in several windows, by placing the command `dev.new()` prior to each set of commands that produces a new graph.

You can close all graphics windows with the `graphics.off()` command.
Aims

- How to present data clearly
- Introducing \textit{ggplot2}

Graphs
- Scatterplots
- Histograms
- Boxplots
- Error bar charts
- Line graphs
We will follow the textbook in using the ggplot package to produce graphs.

Please follow the procedure outlined in Lecture 1 to install and load the ggplot package on the computer you are using now.
In ggplot2 a plot is made up of layers.
The anatomy of a graph
Specifying aesthetics in ggplot2

Aesthetic
- Data
- Colour
- Size
- Shape
- etc.

Specific
- e.g., "Red"
- 2

Variable
- e.g., gender, experimental group

Use aes()
- e.g., `aes(colour = gender), aes(shape = group)`

Don't use aes()
- e.g., `colour = "Red", linetype = 2`

Layer/Geom

Plot
Adding Axis Labels

- R provides the generic function `labels()`, which can be used to add x- and y-axis labels.
- ggplot2 provides the function `labs()`, which does the same thing.
Aims

- How to present data clearly
- Introducing \texttt{ggplot2}

Graphs

- Scatterplots
- Histograms
- Boxplots
- Error bar charts
- Line graphs
Example 1. Narcissism & Facebook (4.4)

□ To run this example, first change your working directory to the directory containing Field’s datasets (M:/jelder/data/field).

□ Now run the following command to read in the data for this example

  facebookData <- read.delim("facebookNarcissism.dat", header=TRUE)
Facebook Narcissism Data

<table>
<thead>
<tr>
<th>id</th>
<th>NPQC_R_Total</th>
<th>Rating_Type</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>31</td>
<td>Attractive</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>31</td>
<td>Fashionable</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>31</td>
<td>Glamourous</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>31</td>
<td>Cool</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>37</td>
<td>Attractive</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>37</td>
<td>Fashionable</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>37</td>
<td>Glamourous</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>37</td>
<td>Cool</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>44.47</td>
<td>Attractive</td>
<td>3</td>
</tr>
</tbody>
</table>
Some Simple ggplot Commands

- First create a graph object, specifying the variables on the x and y axis.
  - `graph <- ggplot(facebookData,aes(NPQC_R_Total, Rating))`
- Now add a scatterplot layer
  - `graph + geom_point()`
- Adjust the shape of the point symbols
  - `graph + geom_point(shape = 17)`
- Adjust the size of the point symbols
  - `graph + geom_point(size = 6)`
- Colour the points according to the rating type
  - `graph + geom_point(aes(colour=Rating_Type))`
Example 2. Exams & Anxiety (4.5)

- To run this example, first change your working directory to the directory containing Field’s datasets (M:/jelder/data/field).

- Now run the following command to read in the data for this example

```r
examData <- read.delim("Exam Anxiety.dat", header=TRUE)
```
Scatterplots: Example 2 (4.5)

- Anxiety and exam performance

- Participants:
  - 103 students

- Measures
  - Time spent revising (hours)
  - Exam performance (%)
  - Exam Anxiety (the EAQ, score out of 100)
  - Gender
## Exam Anxiety Data

<table>
<thead>
<tr>
<th>Code</th>
<th>Revise</th>
<th>Exam</th>
<th>Anxiety</th>
<th>Gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>40</td>
<td>86.298</td>
<td>Male</td>
</tr>
<tr>
<td>2</td>
<td>11</td>
<td>65</td>
<td>88.716</td>
<td>Female</td>
</tr>
<tr>
<td>3</td>
<td>27</td>
<td>80</td>
<td>70.178</td>
<td>Male</td>
</tr>
<tr>
<td>4</td>
<td>53</td>
<td>80</td>
<td>61.312</td>
<td>Male</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>40</td>
<td>89.522</td>
<td>Male</td>
</tr>
<tr>
<td>6</td>
<td>22</td>
<td>70</td>
<td>60.506</td>
<td>Female</td>
</tr>
<tr>
<td>7</td>
<td>16</td>
<td>20</td>
<td>81.462</td>
<td>Female</td>
</tr>
<tr>
<td>8</td>
<td>21</td>
<td>55</td>
<td>75.82</td>
<td>Female</td>
</tr>
<tr>
<td>9</td>
<td>25</td>
<td>50</td>
<td>69.372</td>
<td>Female</td>
</tr>
</tbody>
</table>

...
Simple Scatterplot

```r
scatter <- ggplot(examData, aes(Anxiety, Exam))

scatter + geom_point() + geom_smooth() + labs(x = "Exam Anxiety", y = "Exam Performance %")
```
Simple Scatterplot with Smoother

Scatterplot of exam anxiety against exam performance with a smoother added
Simple Scatterplot With Regression Line

```r
scatter <- ggplot(examData, aes(Anxiety, Exam))

scatter + geom_point() + geom_smooth(method = "lm",
colour = "Red") + labs(x = "Exam Anxiety",
y = "Exam Performance %")
```
Simple Scatterplot

A simple scatterplot with a regression line added.
Grouped Scatterplot

```r
scatter <- ggplot(examData, aes(Anxiety, Exam, colour = Gender))

scatter + geom_point() + geom_smooth(method = "lm", aes(fill = Gender), alpha = 0.1) + labs(x = "Exam Anxiety", y = "Exam Performance %")
```
Grouped Scatterplot

Scatterplot of exam anxiety and exam performance split by gender
Aims

- How to present data clearly
- Introducing \textit{ggplot2}
- Graphs
  - Scatterplots
  - \textbf{Histograms}
  - Boxplots
  - Error bar charts
  - Line graphs
Histograms

- Histograms plot:
  - The score (x-axis)
  - The frequency (y-axis)

- Histograms help us to identify:
  - The shape of the distribution
    - Skew
    - Kurtosis
    - Spread or variation in scores
  - Unusual scores
Histograms: Example

- A biologist was worried about the potential health effects of music festivals.
- Measured the hygiene of 810 concert-goers over the three days of the festival.
- Hygiene was measured using a standardized technique:
  - Score ranged from 0 to 4
    - 0 = you smell like a corpse rotting up a skunk’s arse
    - 4 = you smell of sweet roses on a fresh spring day
To run this example, first change your working directory to the directory containing Field’s datasets (M:/jelder/data/field).

Now run the following command to read in the data for this example

```r
festivalData <- read.delim("DownloadFestival.dat", header=TRUE)
```
<table>
<thead>
<tr>
<th>ticknumb</th>
<th>gender</th>
<th>day1</th>
<th>day2</th>
<th>day3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Male</td>
<td>2.64</td>
<td>1.35</td>
<td>1.61</td>
</tr>
<tr>
<td>2</td>
<td>Female</td>
<td>0.97</td>
<td>1.41</td>
<td>0.29</td>
</tr>
<tr>
<td>3</td>
<td>Male</td>
<td>0.84</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>4</td>
<td>Female</td>
<td>3.03</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>5</td>
<td>Female</td>
<td>0.88</td>
<td>0.08</td>
<td>NA</td>
</tr>
<tr>
<td>6</td>
<td>Male</td>
<td>0.85</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>7</td>
<td>Female</td>
<td>1.56</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>8</td>
<td>Female</td>
<td>3.02</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>9</td>
<td>Male</td>
<td>2.29</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>
Create the plot object:

```r
festivalHistogram <- ggplot(festivalData, aes(day1)) +
                      opts(legend.position = "none")
```

Add the graphical layer:

```r
festivalHistogram + geom_histogram(binwidth = 0.4 ) +
                      labs(x = "Hygiene (Day 1 of Festival)", y = "Frequency")
```
The Resulting Histogram

Hygiene (Day 1 of Festival) vs Frequency
Density Plots

- Histograms show the frequency of values of a variable.
  - The sum over these frequencies is equal to the total number of observations (cases).

- Density plots show the probability density over a random variable.
  - The integral over density $= 1$. 
Density Plot Example

- \( \text{festHist} + \text{geom\_histogram}(\text{aes}(y = \ldots \text{density} \ldots)) + \text{labs}(x = "\text{Hygiene (Day 1 of Festival)}", y = "\text{Density}")) \)

**NB:** some geoms produce new variables. Here `geom\_histogram()` has produced the new variable `density`. We can map an aesthetic to one of these new variables by prepending and appending the variable name with “..”.
Smoothed Density Plot Example

- `festHist + geom_density() + labs(x = "Hygiene (Day 1 of Festival)", y = "Density")`
Boxplots (4.7)

Upper Quartile

Lower Quartile

Median

Top 25%

Middle 50%

Bottom 25%

Female

Gender

Male
Boxplots (Box-Whisker Diagrams)

- Boxplots are made up of a box and two whiskers.
- The box shows:
  - The median
  - The upper and lower quartile
  - The limits within which the middle 50% of scores lie.
- The upper whisker is drawn to the highest score within $1.5 \times IQR$ of the upper box.
- The lower whisker is drawn to the lowest score within $1.5 \times IQR$ of the lower box.
To make a boxplot of the day 1 hygiene scores for males and females, set the variable **Gender** as an aesthetic.

Specify **Gender** to be plotted on the x-axis, and hygiene scores (**day1**) to be plotted on the y-axis:

```r
festivalBoxplot <- ggplot(festivalData, aes(gender, day1))
festivalBoxplot + geom_boxplot() + labs(x = "Gender", y = "Hygiene (Day 1 of Festival)"
```

Note that the order of the bars is determined by the levels attribute of the Gender factor:

```r
levels(festData$gender)
```

[1] "Female" "Male"
The Boxplot

Outlier

Hygiene (Day 1 of Festival)

Gender

Female

Male
Error Bar Charts

- The bar (usually) shows the mean score.
- The error bar sticks out from the bar like a whisker.
- The error bar displays the precision of the mean in one of three ways:
  - The confidence interval (usually 95%)
  - The standard deviation
  - The standard error of the mean
Bar Chart: One Independent Variable

- Is there such a thing as a ‘chick flick’?

- Participants:
  - 20 men
  - 20 women

- Half of each sample saw one of two films:
  - A ‘chick flick’ (*Bridget Jones’s Diary*),
  - Control (*Memento*).

- Outcome measure
  - Physiological arousal as an indicator of how much they enjoyed the film.
To plot the mean arousal score (y-axis) for each film (x-axis) first create the plot object:

```r
bar <- ggplot(chickFlick, aes(film, arousal))
```

To add the mean, displayed as bars, we can add this as a layer to `bar` using the `stat_summary()` function:

```r
bar + stat_summary(fun.y = mean, geom = "bar", fill = "White", colour = "Black")
```
Bar Chart: One Independent Variable

- To add error bars, add these as a layer using `stat_summary()`:
  ```
  + stat_summary(fun.data = mean_cl_normal, geom = "pointrange")
  ```
- Finally, let’s add some nice labels to the graph using `lab()`:
  ```
  + labs(x = "Film", y = "Mean Arousal")
  ```
Bar Chart: One Independent Variable

- If we put all of these commands together we can create the graph by executing the following command:

```r
bar + stat_summary(fun.y = mean, geom = "bar", fill = "White", colour = "Black") + stat_summary(fun.data = mean_cl_normal, geom = "pointrange") + labs(x = "Film", y = "Mean Arousal")
```
Bar Chart: One Independent Variable

Mean Arousal

Film

Bridget Jones' Diary  Memento
Bar Chart: Two Independent Variables

```r
bar <- ggplot(chickFlick, aes(film, arousal, fill = gender))

bar + stat_summary(fun.y = mean, geom = "bar", position="dodge") + stat_summary(fun.data = mean_cl_normal, geom = "errorbar", position = position_dodge(width = 0.90), width = 0.2) + labs(x = "Film", y = "Mean Arousal", fill = "Gender")
```
Bar Chart: Two Independent Variables

- **Gender**:
  - Female
  - Male

- **Films**:
  - Bridget Jones' Diary
  - Memento
Bar Chart: Two Independent Variables

```r
bar <- ggplot(chickFlick, aes(film, arousal, fill = film))

bar + stat_summary(fun.y = mean, geom = "bar") + 
stat_summary(fun.data = mean_cl_normal, geom = "errorbar", width = 0.2) + facet_wrap(~ gender) + 
labs(x = "Film", y = "Mean Arousal") + 
opts(legend.position = "none")
```
Bar Chart: Two Independent Variables
How to cure hiccups?

Participants:
- 15 hiccup sufferers

Each tries four interventions (in random order):
- Baseline
- Tongue-pulling manoeuvres
- Massage of the carotid artery
- Digital rectal massage

Outcome measure
- The number of hiccups in the minute after each procedure
<table>
<thead>
<tr>
<th>Baseline</th>
<th>Tongue</th>
<th>Carotid</th>
<th>Rectum</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>9</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>18</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>17</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>15</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>11</td>
<td>18</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>14</td>
<td>8</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>20</td>
<td>3</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>16</td>
<td>12</td>
<td>3</td>
</tr>
<tr>
<td>17</td>
<td>10</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>19</td>
<td>10</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>14</td>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td>13</td>
<td>22</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
<td>13</td>
<td>4</td>
</tr>
<tr>
<td>14</td>
<td>16</td>
<td>11</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>12</td>
<td>8</td>
<td>3</td>
</tr>
</tbody>
</table>
These data are in the wrong format for ggplot2 to use.

We need all of the scores stacked up in a single column and then another variable that specifies the type of intervention.

We can rearrange the data as follows:

```r
hiccups <- stack(hiccupsData)
names(hiccups) <- c("Hiccups","Intervention")
```
<table>
<thead>
<tr>
<th>Hiccups</th>
<th>Intervention</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>Baseline</td>
</tr>
<tr>
<td>13</td>
<td>Baseline</td>
</tr>
<tr>
<td>9</td>
<td>Baseline</td>
</tr>
<tr>
<td>7</td>
<td>Baseline</td>
</tr>
<tr>
<td>11</td>
<td>Baseline</td>
</tr>
<tr>
<td>14</td>
<td>Baseline</td>
</tr>
<tr>
<td>20</td>
<td>Baseline</td>
</tr>
<tr>
<td>9</td>
<td>Baseline</td>
</tr>
<tr>
<td>17</td>
<td>Baseline</td>
</tr>
<tr>
<td>19</td>
<td>Baseline</td>
</tr>
<tr>
<td>3</td>
<td>Baseline</td>
</tr>
<tr>
<td>13</td>
<td>Baseline</td>
</tr>
<tr>
<td>20</td>
<td>Baseline</td>
</tr>
<tr>
<td>14</td>
<td>Baseline</td>
</tr>
<tr>
<td>13</td>
<td>Baseline</td>
</tr>
<tr>
<td>9</td>
<td>Tongue</td>
</tr>
<tr>
<td>18</td>
<td>Tongue</td>
</tr>
<tr>
<td>17</td>
<td>Tongue</td>
</tr>
<tr>
<td>15</td>
<td>Tongue</td>
</tr>
<tr>
<td>18</td>
<td>Tongue</td>
</tr>
<tr>
<td>8</td>
<td>Tongue</td>
</tr>
<tr>
<td>3</td>
<td>Tongue</td>
</tr>
<tr>
<td>16</td>
<td>Tongue</td>
</tr>
<tr>
<td>10</td>
<td>Tongue</td>
</tr>
<tr>
<td>10</td>
<td>Tongue</td>
</tr>
<tr>
<td>14</td>
<td>Tongue</td>
</tr>
<tr>
<td>22</td>
<td>Tongue</td>
</tr>
<tr>
<td>4</td>
<td>Tongue</td>
</tr>
<tr>
<td>16</td>
<td>Tongue</td>
</tr>
<tr>
<td>12</td>
<td>Tongue</td>
</tr>
<tr>
<td>7</td>
<td>Carotid</td>
</tr>
<tr>
<td>7</td>
<td>Carotid</td>
</tr>
<tr>
<td>7</td>
<td>Carotid</td>
</tr>
<tr>
<td>5</td>
<td>Carotid</td>
</tr>
<tr>
<td>10</td>
<td>Carotid</td>
</tr>
<tr>
<td>10</td>
<td>Carotid</td>
</tr>
</tbody>
</table>
To plot a categorical variable in `ggplot()` it needs to be recognized as a factor:

```r
Hiccups$Intervention_Factor <- factor(hiccups$Intervention, levels = hiccups$Intervention)
```
Line Graphs: One Independent Variable

- We can then create the line graph by executing the following commands:

```r
line <- ggplot(hiccups, aes(Intervention_Factor, Hiccups))

line + stat_summary(fun.y = mean, geom = "point") +
stat_summary(fun.y = mean, geom = "line", aes(group = 1), colour = "Red", linetype = "dashed") +
stat_summary(fun.data = mean_cl_boot, geom = "errorbar", width = 0.2) + labs(x = "Intervention", y = "Mean Number of Hiccups")
```
Line chart with error bars of the mean number of hiccups at baseline and after various interventions
Is text-messaging bad for your grammar?

Participants:
- 50 children

Children split into two groups:
- Text-messaging allowed
- Text-messaging forbidden

Each child measured at two points in time:
- Baseline
- 6 months later

Outcome measure:
- Percentage score on a grammar test
<table>
<thead>
<tr>
<th>Group</th>
<th>Baseline</th>
<th>Six_months</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text Messengers</td>
<td>52</td>
<td>32</td>
</tr>
<tr>
<td>Text Messengers</td>
<td>68</td>
<td>48</td>
</tr>
<tr>
<td>Text Messengers</td>
<td>85</td>
<td>62</td>
</tr>
<tr>
<td>Text Messengers</td>
<td>47</td>
<td>16</td>
</tr>
<tr>
<td>Text Messengers</td>
<td>73</td>
<td>63</td>
</tr>
<tr>
<td>Text Messengers</td>
<td>57</td>
<td>53</td>
</tr>
<tr>
<td>Text Messengers</td>
<td>63</td>
<td>59</td>
</tr>
<tr>
<td>Text Messengers</td>
<td>50</td>
<td>58</td>
</tr>
<tr>
<td>Text Messengers</td>
<td>66</td>
<td>59</td>
</tr>
<tr>
<td>Text Messengers</td>
<td>60</td>
<td>57</td>
</tr>
<tr>
<td>Text Messengers</td>
<td>51</td>
<td>60</td>
</tr>
<tr>
<td>Text Messengers</td>
<td>72</td>
<td>56</td>
</tr>
<tr>
<td>Text Messengers</td>
<td>77</td>
<td>61</td>
</tr>
<tr>
<td>Text Messengers</td>
<td>57</td>
<td>52</td>
</tr>
<tr>
<td>Text Messengers</td>
<td>79</td>
<td>9</td>
</tr>
<tr>
<td>Text Messengers</td>
<td>75</td>
<td>76</td>
</tr>
<tr>
<td>Text Messengers</td>
<td>53</td>
<td>38</td>
</tr>
<tr>
<td>Text Messengers</td>
<td>72</td>
<td>63</td>
</tr>
<tr>
<td>Text Messengers</td>
<td>62</td>
<td>53</td>
</tr>
<tr>
<td>Text Messengers</td>
<td>71</td>
<td>61</td>
</tr>
<tr>
<td>Text Messengers</td>
<td>53</td>
<td>50</td>
</tr>
<tr>
<td>Text Messengers</td>
<td>64</td>
<td>78</td>
</tr>
<tr>
<td>Text Messengers</td>
<td>79</td>
<td>33</td>
</tr>
<tr>
<td>Text Messengers</td>
<td>75</td>
<td>68</td>
</tr>
<tr>
<td>Text Messengers</td>
<td>60</td>
<td>59</td>
</tr>
<tr>
<td>Controls</td>
<td>65</td>
<td>62</td>
</tr>
<tr>
<td>Controls</td>
<td>57</td>
<td>50</td>
</tr>
<tr>
<td>Controls</td>
<td>66</td>
<td>62</td>
</tr>
<tr>
<td>Controls</td>
<td>71</td>
<td>61</td>
</tr>
<tr>
<td>Controls</td>
<td>75</td>
<td>70</td>
</tr>
<tr>
<td>Controls</td>
<td>61</td>
<td>64</td>
</tr>
<tr>
<td>Controls</td>
<td>80</td>
<td>64</td>
</tr>
<tr>
<td>Controls</td>
<td>66</td>
<td>55</td>
</tr>
</tbody>
</table>
Line Graphs for Several Independent Variables

- These data are again in ‘wide’ format but we need the
data to be in ‘long’ format:

```r
textMessages <- melt(textData, id = "Group", measured =
c("Baseline", "Six_months"))
names(textMessages) <- c("Group", "Time",
"Grammar_Score")
```

- We can now change the newly created variable `Time`
so that it is treated as a factor, and provide labels for
the two levels of this variable:

```r
textMessages$Time <- factor(textMessages$Time, levels =
c("Baseline", "Six_months"), labels = c("Baseline", "6 Months"))
```
Line Graphs for Several Independent Variables

```r
line <- ggplot(textMessages, aes(Time, Grammar_Score, colour = Group))

line + stat_summary(fun.y = mean, geom = "point") + stat_summary(fun.y = mean, geom = "line", aes(group = Group)) + stat_summary(fun.data = mean_cl_boot, geom = "errorbar", width = 0.2) + labs(x = "Time", y = "Mean Grammar Score", colour = "Group")
```
Error line graph of the mean grammar score over six months in children who were allowed to text-message versus those who were forbidden.
Saving Your Plots

- You can save your plots as .pdf files using the File- > Save command.

- Hint: When you are going to include a plot in a document, use `+ theme_grey(fontsize)` to adjust the base font size for your plot (for the grey background theme).

- This will allow you to adjust the font size so that the text in the plot is readable even when the plot is reduced in size on the page.
ASSUMPTIONS
Aims

- Assumptions of parametric tests based on the normal distribution
- Understand the assumption of normality
  - Graphical displays
  - Skew
  - Kurtosis
  - Normality tests
- Understand homogeneity of variance
  - Levene’s test
- Know how to correct problems in the data
  - Log, square root and reciprocal transformations
  - Robust tests
Assumptions

- Parametric tests based on the normal distribution assume:
  - Normally distributed
    - Sampling distribution
    - Residuals
  - Homogeneity of variance
  - Interval or ratio level data
  - Independent scores
Assessing Normality

- We don’t have access to the sampling distribution so we usually test the observed data

- Central limit theorem
  - If $N > 30$, the sampling distribution is close to normal anyway

- Graphical displays
  - Q-Q plot (or P-P plot)
  - Histogram

- Values of skew/kurtosis
  - 0 in a normal distribution
  - Convert to $z$ (by dividing value by $SE$)

- Kolmogorov-Smirnov test
  - Tests if data differ from a normal distribution
  - Significant = non-normal data
  - Non-significant = normal data
Histories

- Textbook Errata Page 170
  - `geom_histogram(aes(y = ...density...), colour = "black",...`
    - should, I think, be
  - `geom_density(colour = "black",...`
Normality Example

- A biologist was worried about the potential health effects of music festivals.
- Download Music Festival
- Measured the hygiene of 810 concert-goers over the three days of the festival.
- Hygiene was measured using a standardized technique:
  - Score ranged from 0 to 4
Q-Q Plots

- Q-Q (Quantile-Quantile) plots show a scatter plot of percentiles scores of the empirical distribution vs some model distribution (typically normal).
- For example, suppose you have an empirical distribution with \( n = 1,000 \) scores.
- Sort the scores in increasing order.
- The first score now corresponds to a percentile of \( 1/1000 \), the second score corresponds to \( 2/1000 \), etc…
- Thus the first \( x \)-value on the scatterplot will be \( 1/1000 \), the second will be \( 2/1000 \) etc…
Q-Q Plots

- Now consider the standard normal distribution (mean = 0, SD = 1).
- The first y value satisfies \( \text{cdf}(y) = 1/1000 \) where \( \text{cdf} \) is the cumulative density of the standard normal.
- The second y value satisfies \( \text{cdf}(y) = 2/1000 \), etc…
- If the empirical distribution is normal, the resulting scatterplot should form a straight line.
The Q-Q Plot

- To draw a Q-Q plot of the hygiene scores for day 1 of the music festival:
  
  ```
  qqplot.day1 <- qplot(sample = dlf$day1, stat="qq")
  qqplot.day1
  ```

- To draw a Q-Q plot of the hygiene scores for day 2 of the music festival:
  
  ```
  qqplot.day2 <- qplot(sample = dlf$day2, stat="qq")
  qqplot.day2
  ```
The Q-Q Plot

Hygiene Scores: Day 1

Hygiene Scores: Day 2

Normal

Not Normal
Quantifying Normality

- We can complement these visual methods with quantitative measures of deviations from normality.

- Method 1:
  ```
  install.packages("psych")
  library(psych)
  describe(dlf$day1)
  ```

- Method 2:
  ```
  install.packages("pastecs")
  library(pastecs)
  stat.desc(dlf$day1, basic = FALSE, norm = TRUE)
  ```
Assessing Skew and Kurtosis

<table>
<thead>
<tr>
<th></th>
<th>day1</th>
<th>day2</th>
<th>day3</th>
</tr>
</thead>
<tbody>
<tr>
<td>median</td>
<td>1.7900000000</td>
<td>7.900000e-01</td>
<td>7.600000e-01</td>
</tr>
<tr>
<td>mean</td>
<td>1.770828183</td>
<td>9.609091e-01</td>
<td>9.765041e-01</td>
</tr>
<tr>
<td>SE.mean</td>
<td>0.024396670</td>
<td>4.436095e-02</td>
<td>6.404352e-02</td>
</tr>
<tr>
<td>CI.mean.0.95</td>
<td>0.047888328</td>
<td>8.734781e-02</td>
<td>1.267805e-01</td>
</tr>
<tr>
<td>var</td>
<td>0.481514784</td>
<td>5.195239e-01</td>
<td>5.044934e-01</td>
</tr>
<tr>
<td>std.dev</td>
<td>0.693912663</td>
<td>7.207801e-01</td>
<td>7.102770e-01</td>
</tr>
<tr>
<td>coef.var</td>
<td>0.391857702</td>
<td>7.501022e-01</td>
<td>7.273672e-01</td>
</tr>
<tr>
<td>skewness</td>
<td>-0.003155393</td>
<td>1.082811e+00</td>
<td>1.007813e+00</td>
</tr>
<tr>
<td>skew.2SE</td>
<td>-0.018353763</td>
<td>3.611574e+00</td>
<td>2.309035e+00</td>
</tr>
<tr>
<td>kurtosis</td>
<td>-0.423991408</td>
<td>7.554615e-01</td>
<td>5.945454e-01</td>
</tr>
<tr>
<td>kurt.2SE</td>
<td>-1.234611514</td>
<td>1.264508e+00</td>
<td>6.862946e-01</td>
</tr>
<tr>
<td>(SD / Mean)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>normtest.W</td>
<td>0.995907247</td>
<td>9.083185e-01</td>
<td>9.077513e-01</td>
</tr>
<tr>
<td>(p-value for S-W Test) normtest.p</td>
<td>0.031846386</td>
<td>1.281495e-11</td>
<td>3.804334e-07</td>
</tr>
</tbody>
</table>
We can also use `describe()` and `stat.desc()` with more than one variable at the same time using `cbind()`:

```r
describe(cbind(dlf$day1, dlf$day2, dlf$day3))
```

```r
stat.desc(cbind(dlf$day1, dlf$day2, dlf$day3), basic = FALSE, norm = TRUE)
```
End of Lecture

Sept 28, 2012
Another Example

- Performance on statistics exam
- Participants
  - $N = 100$ students
- Measures
  - Exam: first-year exam scores as a percentage
  - Computer: measure of computer literacy, %
  - Lecture: percentage of lectures attended
  - Numeracy: a measure of numerical ability out of 15
  - Uni: whether the student attended Sussex University or Duncetown University
### Data

<table>
<thead>
<tr>
<th>exam</th>
<th>computer</th>
<th>lectures</th>
<th>numeracy</th>
<th>uni</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>54</td>
<td>75</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>47</td>
<td>8.5</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>40</td>
<td>58</td>
<td>69.5</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>37</td>
<td>67</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>40</td>
<td>53</td>
<td>44.5</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

...
Q-Q Plots

First Year Exam Scores

Numeracy

Normal Q-Q plots of R exam scores and numeracy show deviations from normality
Running Analysis by Group

- Suppose we wish to get separate statistics for each of the universities.
- To do this we can use the `by()` function:
  - `by(data = dataFrame or matrix, INDICES = grouping variable, FUN = a function that you want to apply to the data)`
- Example:
  - `by(data = rexam$exam, INDICES = rexam$uni, FUN = describe)`
- You can pass options to the function you are using by adding them at the end. For example,
  - `by(rexam$exam, rexam$uni, stat.desc, basic = FALSE, norm = TRUE)`
Assessing Normality

- Shapiro-Wilk test for exam and numeracy for whole sample:
  - `shapiro.test(rexam$exam)`
  - `shapiro.test(rexam$numeracy)`

- Output:
  ```
  > Shapiro-Wilk normality test
data: rexam$exam
W = 0.9613, p-value = 0.004991

> Shapiro-Wilk normality test
data: rexam$numeracy
W = 0.9244, p-value = 2.424e-05
  ```
Assessing Normality

- Shapiro-Wilk test for exam and numeracy split by university:

  by(rexam$exam, rexam$uni, shapiro.test)
  by(rexam$numeracy, rexam$uni, shapiro.test)
Assessing Normality

- Output for exam:
  rexam$uni: Duncetown University
  Shapiro-Wilk normality test
  data:  dd[x, ]
  W = 0.9722, p-value = 0.2829

- rexam$uni: Sussex University
  Shapiro-Wilk normality test
  data:  dd[x, ]
  W = 0.9837, p-value = 0.7151
Assessing Normality

Output for numeracy:

rexam$uni: Duncetown University

Shapiro-Wilk normality test

data:  dd[x, ]
W = 0.9408, p-value = 0.01451

-------------------------------------------------------------------

rexam$uni: Sussex University

Shapiro-Wilk normality test

data:  dd[x, ]
W = 0.9323, p-value = 0.006787
Homogeneity of Variance

- Homogeneity of variance means that the variance in the dependent variable does not change as a function of the independent variable(s).

- Most inferential statistics (e.g., t-tests, ANOVAs) are only valid if homogeneity of variance applies.
Assessing Homogeneity of Variance

- **Graphs**
- **Levene’s test**
  - Tests if variances in different groups are the same.
  - Significant = variances not equal
  - Non-significant = variances are equal
- **Variance ratio**
  - With 2 or more groups
  - VR = largest variance/smallest variance
  - If VR < 2, homogeneity can be assumed.
Homogeneity of Variance

Homogeneous

Heterogeneous
Assessing Homogeneity of Variance with R Commander

![Image of R Commander interface with Levene's Test]

- **Groups (pick one):** uni
- **Response Variable (pick one):** exam

- **Center:** median

![Image of Levene's Test dialog box]

The Levene's Test dialog box is used to assess homogeneity of variance. The groups are set to 'uni' and the response variable is set to 'exam'. The center is set to median.
Assessing Homogeneity of Variance with R

- Use the `leveneTest()` function from the `car` package:
  ```r
  leveneTest(outcome variable, group, center = median/mean)
  ```

- Levene’s test for the exam and numeracy scores:
  ```r
  leveneTest(rexam$exam, rexam$uni)
  leveneTest(rexam$numeracy, rexam$uni)
  ```
Output for Levene’s Test

> leveneTest(rexam$exam, rexam$uni)

Levene's Test for Homogeneity of Variance (center = median)

<table>
<thead>
<tr>
<th></th>
<th>Df</th>
<th>F value</th>
<th>Pr(&gt;F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>group 1</td>
<td>1</td>
<td>2.0886</td>
<td>0.1516</td>
</tr>
<tr>
<td></td>
<td>98</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

> leveneTest(rexam$numeracy, rexam$uni)

Levene's Test for Homogeneity of Variance (center = median)

<table>
<thead>
<tr>
<th></th>
<th>Df</th>
<th>F value</th>
<th>Pr(&gt;F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>group 1</td>
<td>1</td>
<td>5.366</td>
<td>0.02262 *</td>
</tr>
<tr>
<td></td>
<td>98</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Correcting Data Problems

- **Log transformation** ($\log(X_i)$):
  - Reduce positive skew.

- **Square root transformation** ($\sqrt{X_i}$):
  - Also reduces positive skew.

- **Reciprocal transformation** ($1/X_i$):
  - Dividing 1 by each score also reduces the impact of large scores. This transformation reverses the scores; you can avoid this by reversing the scores before the transformation, $1/(X_{\text{Highest}} - X_i)$. 

Correcting Data Problems

- Log transformation:
  ```r
dlfs$\text{logday1} <- \text{log}(dlf$\text{day1})
  ```

- Square root transformation:
  ```r
dlfs$\text{sqrtday1} <- \text{sqrt}(\text{day1})
  ```

- Reciprocal transformation:
  ```r
dlfs$\text{recday1} <- 1/(dlf$\text{day1})
  ```
Example: Log Transform

- **Log Transform of Hygiene Score on Day 2**
- **Density Distribution**
Example: Square Root Transform

Hygiene score on Day 2

Square Root of Hygiene score on Day 2
What do we do when our assumptions fail?

- Trimmed means
- M-estimators
- Bootstrapping
Aims

- Assumptions of parametric tests based on the normal distribution
- Understand the assumption of normality
  - Graphical displays
  - Skew
  - Kurtosis
  - Normality tests
- Understand homogeneity of variance
  - Levene’s test
- Know how to correct problems in the data
  - Log, square root and reciprocal transformations
  - Pitfalls and alternatives
  - Robust tests