Describing data

Dr. Jeffrey Strickland

8/30/2018

Describing Plant Growth

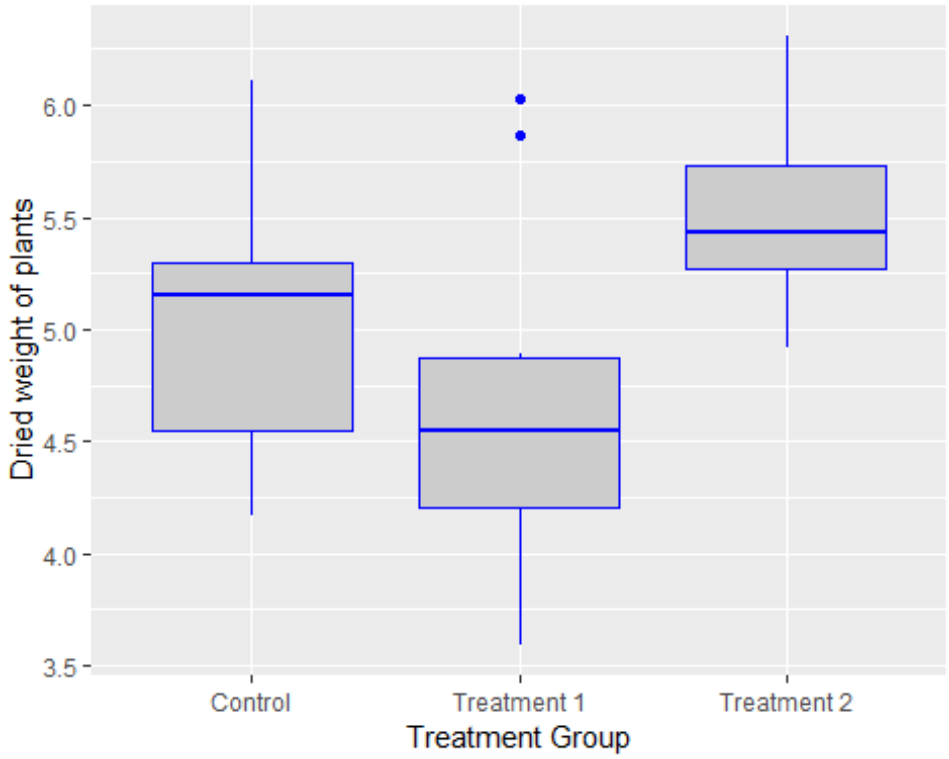
As an example we consider one of the data sets available with R relating to an experiment into plant growth. The purpose of the experiment was to compare the yields on the plants for a control group and two treatments of interest. The response variable was a measurement taken on the dried weight of the plants.

The first step in the investigation is to take a copy of the data frame so that we can make some adjustments as necessary while leaving the original data alone. We use the factor function to re-define the labels of the group variables that will appear in the output and graphs:

```
plant.df = PlantGrowth
plant.df$group = factor(plant.df$group,
  labels = c("Control", "Treatment 1", "Treatment 2"))
```

The labels argument is a list of names corresponding to the levels of the group factor variable. A boxplot of the distributions of the dried weights for the three competing groups is created using the ggplot package:

```
require(ggplot2)

## Loading required package: ggplot2

ggplot(plant.df, aes(x = group, y = weight)) +
  geom_boxplot(fill = "grey80", colour = "blue") +
  scale_x_discrete() + xlab("Treatment Group") +
  ylab("Dried weight of plants")
```



The geom_boxplot() option is used to specify background and outline colours for the boxes. The axis labels are created with the xlab() and ylab() options. The plot that is produced looks like this:

Initial inspection of the data suggests that there are differences in the dried weight for the two treatments but it is not so clear cut to determine whether the treatments are different to the control group. To investigate these differences we fit the one-way ANOVA model using the lm function and look at the parameter estimates and standard errors for the treatment effects. The function call is:

```
plant.mod1 = lm(weight ~ group, data = plant.df)
```

We save the model fitted to the data in an object so that we can undertake various actions to study the goodness of the fit to the data and other model assumptions. The standard summary of a lm object is used to produce the following output: summary(plant.mod1)

The model output indicates some evidence of a difference in the average growth for the 2nd treatment compared to the control group. An analysis of variance table for this model can be produced via the anova command:

```
anova(plant.mod1)

## Analysis of Variance Table
##
## Response: weight
##           Df  Sum Sq Mean Sq F value  Pr(>F)
## group      2  3.7663  1.8832  4.8461 0.01591 *
```

```
## Residuals 27 10.4921  0.3886
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This table confirms that there are differences between the groups which were highlighted in the model summary. The function confint is used to calculate confidence intervals on the treatment parameters, by default 95% confidence intervals:

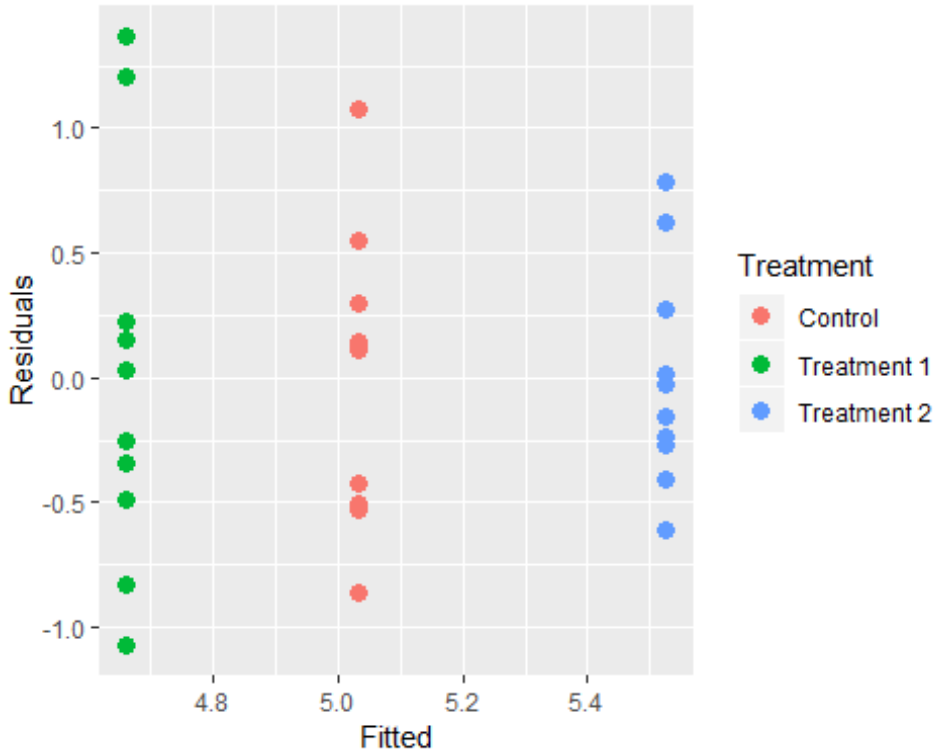
```
confint(plant.mod1)

##                          2.5 %    97.5 %
## (Intercept)       4.62752600 5.4364740
## groupTreatment 1 -0.94301261 0.2010126
## groupTreatment 2 -0.07801261 1.0660126
```

The model residuals can be plotted against the fitted values to investigate the model assumptions. First we create a data frame with the fitted values, residuals and treatment identifiers:

```
plant.mod = data.frame(Fitted = fitted(plant.mod1),
  Residuals = resid(plant.mod1), Treatment = plant.df$group)
```

and then produce the plot:

```
ggplot(plant.mod, aes(Fitted, Residuals, colour = Treatment)) +
geom_point(size=3)
```



We can see that there is no major problem with the diagnostic plot but some evidence of different variabilities in the spread of the residuals for the three treatment groups. The r function aov builds an ANOVA model from the data rather than from a model.

```
plant.aov<-aov(weight ~ group,plant.df)
```

The basic result doen not give a great deal of information. We need to view the summary so try:

```
summary(plant.aov)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## group        2  3.766  1.8832   4.846 0.0159 *
## Residuals   27 10.492  0.3886
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

So far we have conducted a simple one-way anova. In this instance we see that there is a significant effect of diet upon growth. However, there are 6 treatments. We would like to know which of these treatments are significantly different from the controls and from other treatments. We need a post-hoc test. R provides a simple function to carry out the Tukey HSD test.

This will show all the paired comparisons like so:

```
TukeyHSD(plant.aov)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = weight ~ group, data = plant.df)
##
## $group
##                           diff        lwr       upr     p adj
## Treatment 1-Control     -0.371 -1.0622161 0.3202161 0.3908711
## Treatment 2-Control      0.494 -0.1972161 1.1852161 0.1979960
## Treatment 2-Treatment 1  0.865  0.1737839 1.5562161 0.0120064
```

The table/output shows us the difference between pairs, the 95% confidence interval(s) and the p-value of the pairwise comparisons. All we need to know!

Analysis of Vatiance

The analysis of variance (ANOVA) model can be extended from making a comparison between multiple groups to take into account additional factors in an experiment. The simplest extension is from one-way to two-way ANOVA where a second factor is included in the model as well as a potential interaction between the two factors.

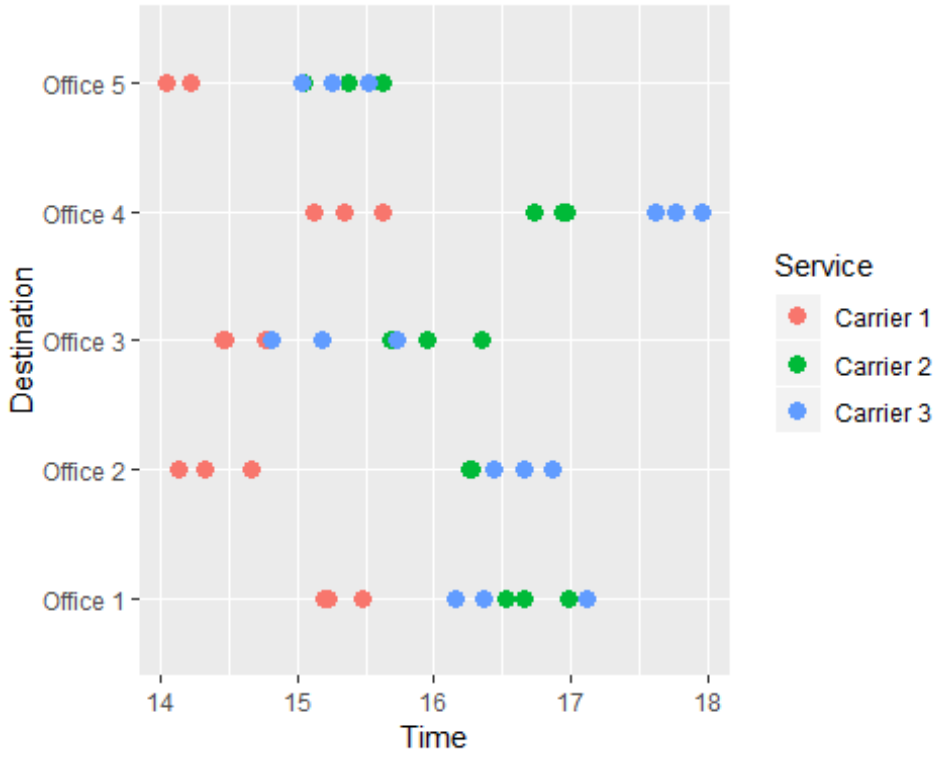
As an example consider a company that regularly has to ship parcels between its various (five for this example) sub-offices and has the option of using three competing parcel delivery services, all of which charge roughly similar amounts for each delivery. To

determine which service to use, the company decides to run an experiment shipping three packages from its head office to each of the five sub-offices. The delivery time for each package is recorded and the data loaded into R:

```
delivery.df = data.frame(
  Service = c(rep("Carrier 1", 15), rep("Carrier 2", 15),
    rep("Carrier 3", 15)),
  Destination = c(rep(c("Office 1", "Office 2", "Office 3",
    "Office 4", "Office 5"), 9)),
  Time = c(15.23, 14.32, 14.77, 15.12, 14.05,
  15.48, 14.13, 14.46, 15.62, 14.23, 15.19, 14.67, 14.48, 15.34, 14.22,
  16.66, 16.27, 16.35, 16.93, 15.05, 16.98, 16.43, 15.95, 16.73, 15.62,
  16.53, 16.26, 15.69, 16.97, 15.37, 17.12, 16.65, 15.73, 17.77, 15.52,
  16.15, 16.86, 15.18, 17.96, 15.26, 16.36, 16.44, 14.82, 17.62, 15.04)
)
```

The data is then displayed using a dot plot for an initial visual investigation of any trends in delivery time between the three services and across the five sub-offices. The colour aesthetic is used to distinguish between the three services in the plot.

```
ggplot(delivery.df, aes(Time, Destination, colour = Service)) +
geom_point(size=3)
```



The graph shows a general pattern of service carrier 1 having shorter delivery times than the other two services. There is also an indication that the differences between the services varies for the five sub-offices and we might expect the interaction term to be significant in the two-way ANOVA model. To fit the two-way ANOVA model we use this code:

```
delivery.mod1 = aov(Time ~ Destination*Service, data = delivery.df)
```

The * symbol instructs R to create a formula that includes main effects for both Destination and Service as well as the two-way interaction between these two factors. We save the fitted model to an object which we can summarise as follows to test for importance of the various model terms:

```
summary(delivery.mod1)
```

```
##                      Df Sum Sq Mean Sq F value   Pr(>F)
## Destination           4 17.542   4.385  61.155 5.41e-14 ***
## Service               2 23.171  11.585 161.560  < 2e-16 ***
## Destination:Service   8  4.189   0.524   7.302 2.36e-05 ***
## Residuals            30  2.151   0.072
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

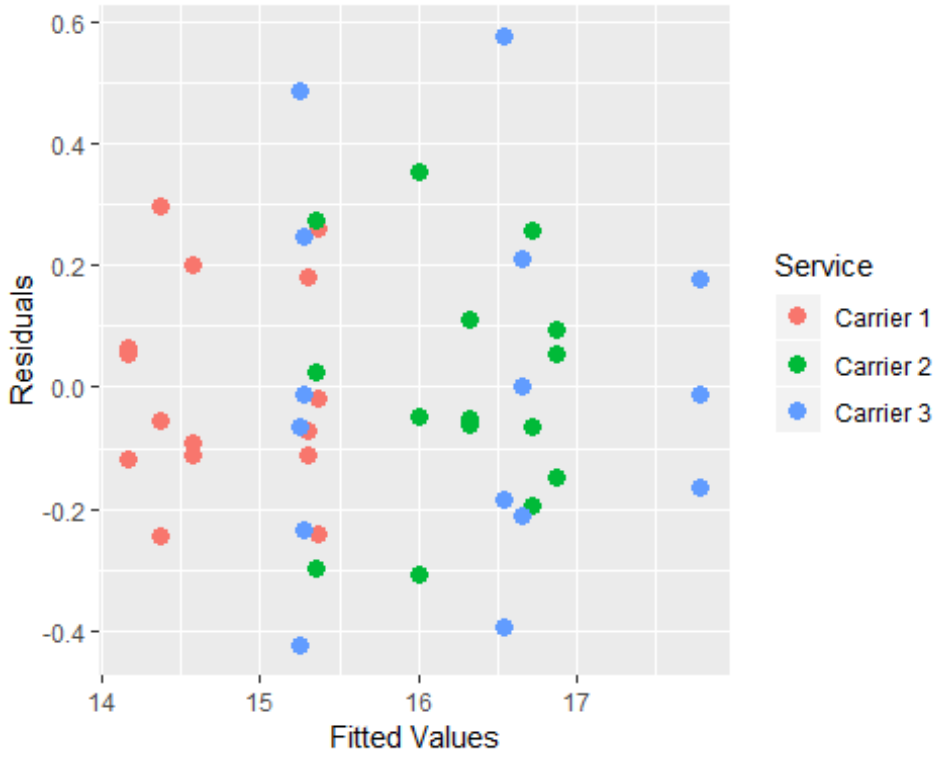
We have strong evidence here that there are differences between the three delivery services, between the five sub-office destinations and that there is an interaction between destination and service in line with what we saw in the original plot of the data. Now that we have fitted the model and identified the important factors we need to investigate the model diagnostics to ensure that the various assumptions are broadly valid.

We can plot the model residuals against fitted values to look for obvious trends that are not consistent with the model assumptions about independence and common variance. The first step is to create a data frame with the fitted values and residuals from the above model:

```
delivery.res = delivery.df
delivery.res$M1.Fit = fitted(delivery.mod1)
delivery.res$M1.Resid = resid(delivery.mod1)
```

Then a scatter plot is used to display the fitted values and residuals where the colour asthetic highlights which points correspond to the three competing delivery services:

```
ggplot(delivery.res, aes(M1.Fit, M1.Resid, colour = Service)) +
geom_point(size=3) +
  xlab("Fitted Values") + ylab("Residuals")
```

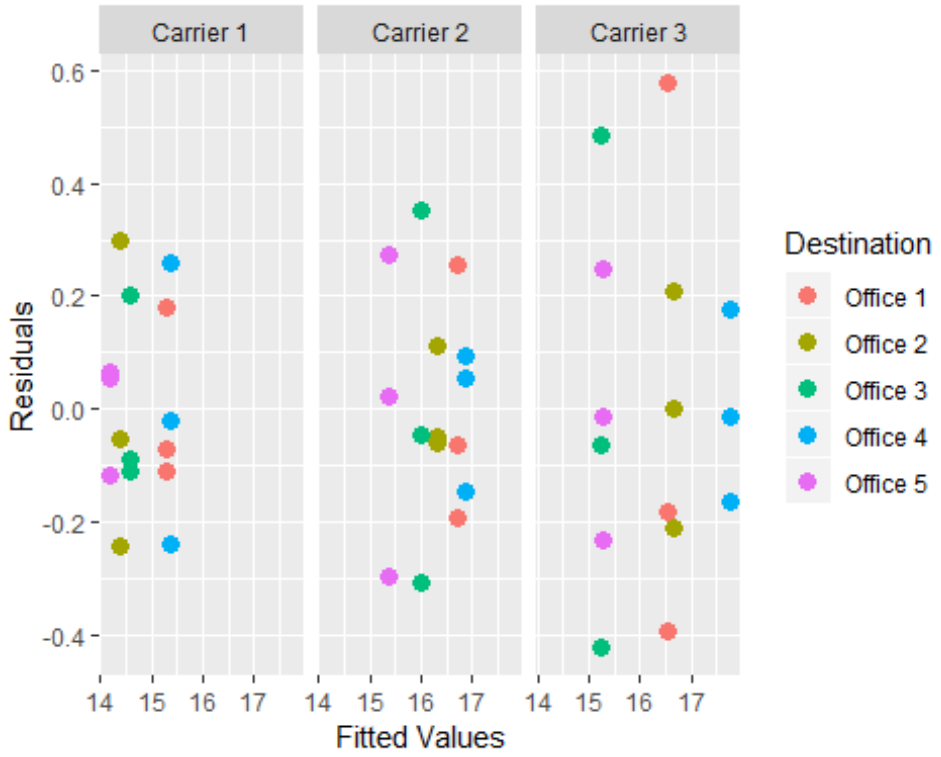


There are no obvious patterns in this plot that suggest problems with the two-way ANOVA model that we fitted to the data. As an alternative display we could separate the residuals into destination sub-offices, where the facet_wrap() function instructs ggplot to create a separate display (panel) for each of the destinations.

```
ggplot(delivery.res, aes(M1.Fit, M1.Resid, colour = Service)) +
  geom_point(size=3) + xlab("Fitted Values") + ylab("Residuals") +
  facet_wrap( ~ Destination)
```

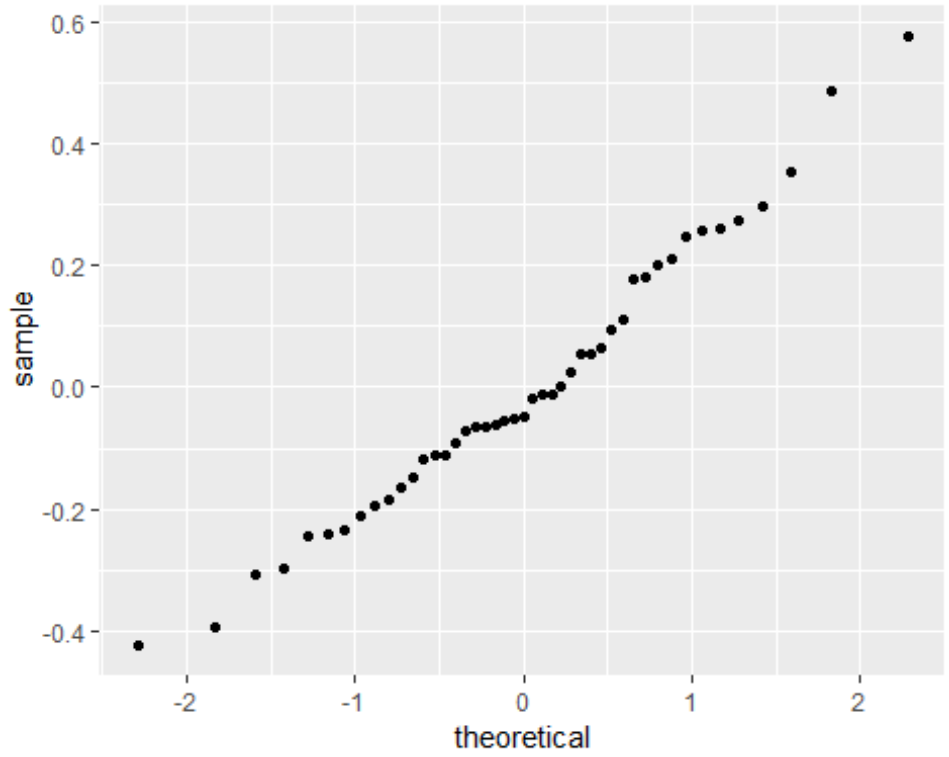
No obvious problems in this diagnostic plot. We could also consider dividing the data by delivery service to get a different view of the residuals:

```
ggplot(delivery.res, aes(M1.Fit, M1.Resid, colour = Destination)) +
  geom_point(size=3) + xlab("Fitted Values") + ylab("Residuals") +
  facet_wrap( ~ Service)
```

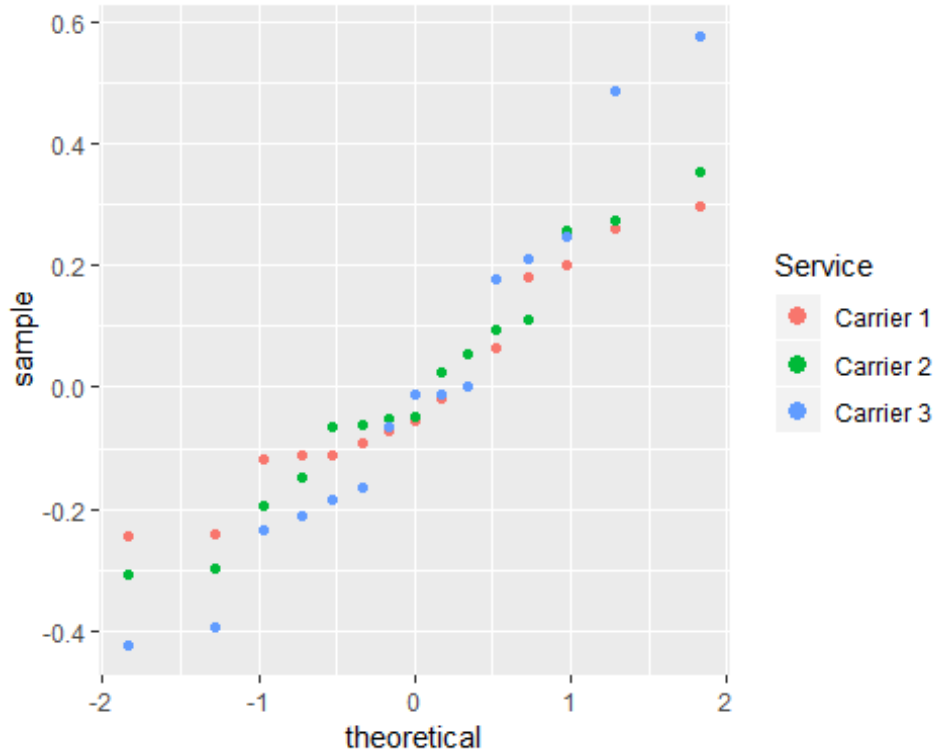


Again there is nothing substantial here to lead us to consider an alternative analysis. Lastly we consider the normal probability plot of the model residuals, using the stat_qq() option:

```
ggplot(delivery.res, aes(sample = M1.Resid)) + stat_qq()
```




```
ggplot(delivery.res, aes(sample = M1.Resid, colour=Service)) + stat_qq() +
  geom_point(x=delivery.res$M1.Fit, y=delivery.res$M1.Resid, size=3)
```



This plot is very close to the straight line we would expect to observe if the data was a close approximation to a normal distribution. To round off the analysis we look at the Tukey HSD multiple comparisons to confirm that the differences are between delivery service 1 and the other two competing services:

```
TukeyHSD(delivery.mod1, which = "Service")
```

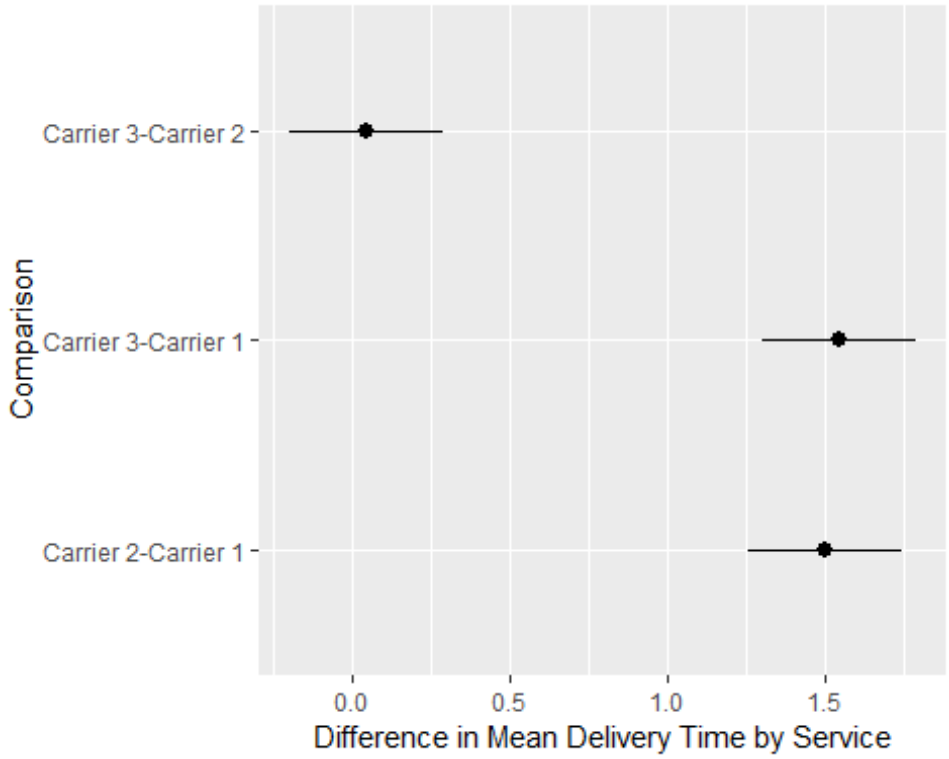
```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Time ~ Destination * Service, data = delivery.df)
##
## $Service
##                          diff        lwr       upr     p adj
## Carrier 2-Carrier 1 1.498667  1.2576092 1.7397241 0.0000000
## Carrier 3-Carrier 1 1.544667  1.3036092 1.7857241 0.0000000
## Carrier 3-Carrier 2 0.046000 -0.1950575 0.2870575 0.8856246
```

Even with the multiple comparison post-hoc adjustment there is very strong evidence for the differences that we have consistenly observed throughout the analysis. We can use ggplot to visualise the difference in mean delivery time for the services and the 95% confidence intervals on these differences. We create a data frame from the TukeyHSD output by extracting the component relating to the delivery service comparison and add the text labels by extracting the row names from the data frame.

```
delivery.hsd = data.frame(TukeyHSD(delivery.mod1, which = "Service")$Service)
delivery.hsd$Comparison = row.names(delivery.hsd)
```

We then use the geom_pointrange() to specify lower, middle and upper values based on the three pairwise comparisons of interest.

```
ggplot(delivery.hsd, aes(Comparison, y = diff, ymin = lwr, ymax = upr)) +
  geom_pointrange() + ylab("Difference in Mean Delivery Time by Service") +
  coord_flip()
```



The coord_flip() is used to make the confidence intervals horizontal rather than vertical on the graph.

Experimental Designs

This example requires the R stats package. There are three groups with seven observations per group. We denote group i values by y_i:

```
y1 = c(18.2, 20.1, 17.6, 16.8, 18.8, 19.7, 19.1)
y2 = c(17.4, 18.7, 19.1, 16.4, 15.9, 18.4, 17.7)
y3 = c(15.2, 18.8, 17.7, 16.5, 15.9, 17.1, 16.7)

local({pkg <- select.list(sort(.packages(all.available =  TRUE)),
graphics=TRUE)
if(nchar(pkg)) library(pkg, character.only=TRUE)})
```

Now we combine them into one long vector, with a second vector, group, identifying group membership:

```
y = c(y1, y2, y3)
n = rep(7, 3)
n

## [1] 7 7 7

group = rep(1:3, n)
group
```

```
##  [1] 1 1 1 1 1 1 1 2 2 2 2 2 2 2 3 3 3 3 3 3 3
```

Here are summaries by group and for the combined data. First we show stem-leaf diagrams.

```
tmp = tapply(y, group, stem)
```

```
##
##   The decimal point is at the |
##
##   16 | 8
##   17 | 6
##   18 | 28
##   19 | 17
##   20 | 1
##
##
##   The decimal point is at the |
##
##   15 | 9
##   16 | 4
##   17 | 47
##   18 | 47
##   19 | 1
##
##
##   The decimal point is at the |
##
##   15 | 29
##   16 | 57
##   17 | 17
##   18 | 8
```

```
stem(y)
```

```
##
##   The decimal point is at the |
##
##   15 | 299
##   16 | 4578
##   17 | 14677
##   18 | 24788
##   19 | 117
##   20 | 1
```

Now we show summary statistics by group and overall. We locally define a temporary function, tmpfn, to make this easier.

```
tmpfn = function(x) c(sum = sum(x), mean = mean(x), var = var(x),n =
length(x))
tapply(y, group, tmpfn)
```

```
## $`1`
##        sum       mean        var          n
## 130.300000  18.614286   1.358095   7.000000
##
## $`2`
##        sum       mean        var          n
## 123.600000  17.657143   1.409524   7.000000
##
## $`3`
##        sum       mean        var          n
## 117.900000  16.842857   1.392857   7.000000
```

```
tmpfn(y)
```

```
##        sum       mean        var          n
## 371.800000  17.704762   1.798476  21.000000
```

```
data = data.frame(y = y, group = factor(group))
fit = lm(y ~ group, data)
anova(fit)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value  Pr(>F)
## group      2 11.007  5.5033  3.9683 0.03735 *
## Residuals 18 24.963  1.3868
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
df = anova(fit)[, "Df"]
names(df) = c("trt", "err")
```

Get F Values: First we extract the treatment and error degrees of freedom. Then we use qt to get the tabled F values.

```
df
```

```
## trt err
##   2  18
```

```
alpha = c(0.05, 0.01)
qf(alpha, df["trt"], df["err"], lower.tail = FALSE)
```

```
## [1] 3.554557 6.012905
```

A confidence interval on the pooled variance can be computed as well using the anova(fit) object. First we get the residual sum of squares, SSTrt, then we divide by the appropriate chi-square tabled values.

```
anova(fit)["Residuals", "Sum Sq"]
```

```
## [1] 24.96286
```

```
anova(fit)["Residuals", "Sum Sq"]/qchisq(c(0.025, 0.975), 18)
```

```
## [1] 3.0328790 0.7918086
```

```
anova(fit)["Residuals", "Sum Sq"]/qchisq(c(0.025, 0.975), 18,lower.tail =
FALSE)
```

```
## [1] 0.7918086 3.0328790
```

Motivational Questions

1. In supervised machine learning, is it necessary to perform data descriptive analysis?
2. In unsupervised machine learning, is it necessary to perform data descriptive analysis?
3. Why is there a difference?