Introduction to R

Session 5 – Advanced Graphics

Statistical Consulting Centre

20 July, 2017

1. T-test

1. Do a two-sample *t*-test (two-sided) to see whether `Mercury` for acidic lakes is different from other `pHtype` levels, assuming

(a) population variances are equal, and

```
joined.long.df$pHtypeAcidic <-  with(joined.long.df, ifelse(pH < 7, "acidity", "Others"))

with(joined.long.df, t.test(Mercury~pHtypeAcidic, var.equal = TRUE))
```

```
##
##  Two Sample t-test
##
## data:  Mercury by pHtypeAcidic
## t = 7.6143, df = 157, p-value = 2.323e-12
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.2749195 0.4675110
## sample estimates:
## mean in group acidity  mean in group Others
##             0.6761464             0.3049312
```

(b) population variances are not equal

```
with(joined.long.df, t.test(Mercury~pHtypeAcidic, var.equal = FALSE))
```

```
##
##  Welch Two Sample t-test
##
## data:  Mercury by pHtypeAcidic
## t = 7.9005, df = 154.55, p-value = 4.852e-13
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.2783974 0.4640330
## sample estimates:
## mean in group acidity  mean in group Others
##             0.6761464             0.3049312
```

Hint: Create a new variable `pHtypeAcidic` for just acidic group.

2. How do the results from using the following code compare with the results you got in question 1?

```
acid.mer <- with(joined.long.df, mercury[pHtype == "acidity"])
not.acid.mer <- with(joined.long.df, mercury[pHtype != "acidity"])

t.test(acid.mer, not.acid.mer)
```

```r
acid.mer <- with(joined.long.df, Mercury[pHtype == "acidity"])
not.acid.mer <- with(joined.long.df, Mercury[pHtype != "acidity"])

t.test(acid.mer, not.acid.mer)
```

```
##
##  Welch Two Sample t-test
##
## data:  acid.mer and not.acid.mer
## t = 7.9005, df = 154.55, p-value = 4.852e-13
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.2783974 0.4640330
## sample estimates:
## mean of x mean of y
## 0.6761464 0.3049312
```

2. One-way ANOVA

1. Perform an ANOVA to test the null hypothesis that the mean mercury concentrations are all equal for all three Calcium levels.

```r
myaov <- with(joined.long.df, aov(Mercury ~ Calcium))
```

2. Use `summary()` to check the overall signifcance of `Calcium`.

```r
summary(myaov)
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)
## Calcium       2  4.226  2.1131   21.27 6.79e-09 ***
## Residuals   156 15.499  0.0994
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3. Use `model.tables()` to calculate the mean mercury concentrations for each Calcium level.

```r
model.tables(myaov, type = "means")
```

```
## Tables of means
## Grand mean
##
## 0.5220571
##
##  Calcium
##         Low  Medium    High
##      0.7226  0.5057  0.3159
## rep 54.0000 57.0000 48.0000
```

4. Carry out pairwise comparisons of mean mercury concentrations between pairs of Calcium levels, adjusting the p-values using Tukey's Honestly Signifcance Difference.

```r
TukeyHSD(myaov)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
```

```
## Fit: aov(formula = Mercury ~ Calcium)
##
## $Calcium
##                   diff        lwr         upr     p adj
## Medium-Low  -0.2168716 -0.3585110 -0.07523211 0.0011392
## High-Low    -0.4066605 -0.5546189 -0.25870211 0.0000000
## High-Medium -0.1897890 -0.3359036 -0.04367433 0.0070127
```

3. Two-way ANOVA

1. Perform an Two-way ANOVA to test the null hypothesis that the mean mercury concentrations are all equal for all three Calcium levels across all three days.

```
myaov <- with(joined.long.df, aov(Mercury ~ Calcium*Time))
```

2. Use `summary()` to check the overall signifcance.

```
summary(myaov)
```

```
##                Df Sum Sq Mean Sq F value   Pr(>F)
## Calcium         2  4.226  2.1131  20.522 1.32e-08 ***
## Time            2  0.017  0.0086   0.084    0.920
## Calcium:Time    4  0.037  0.0092   0.089    0.986
## Residuals     150 15.445  0.1030
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3. Use `model.tables()` to calculate the mean mercury concentrations for each Calcium level.

```
model.tables(myaov, type = "means")
```

```
## Tables of means
## Grand mean
##
## 0.5220571
##
##  Calcium
##         Low  Medium    High
##      0.7226  0.5057  0.3159
## rep 54.0000 57.0000 48.0000
##
##  Time
##        Day1    Day2    Day3
##      0.5087  0.5233  0.5342
## rep 53.0000 53.0000 53.0000
##
##  Calcium:Time
##         Time
## Calcium  Day1   Day2   Day3
##   Low     0.706  0.737  0.725
##   rep    18.000 18.000 18.000
##   Medium  0.478  0.496  0.543
##   rep    19.000 19.000 19.000
##   High    0.323  0.315  0.309
##   rep    16.000 16.000 16.000
```

4. Carry out pairwise comparisons of mean mercury concentrations, adjusting the p-values using Tukey's Honestly Signifcance Difference.

```r
TukeyHSD(myaov)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Mercury ~ Calcium * Time)
##
## $Calcium
##                   diff        lwr        upr     p adj
## Medium-Low  -0.2168716 -0.3611196 -0.0726235 0.0014408
## High-Low    -0.4066605 -0.5573439 -0.2559771 0.0000000
## High-Medium -0.1897890 -0.3385946 -0.0409833 0.0083306
##
## $Time
##                 diff        lwr       upr     p adj
## Day2-Day1 0.01455126 -0.1330058 0.1621083 0.9704115
## Day3-Day1 0.02544596 -0.1221111 0.1730030 0.9122994
## Day3-Day2 0.01089469 -0.1366623 0.1584517 0.9833019
##
## $`Calcium:Time`
##                                diff         lwr         upr     p adj
## Medium:Day1-Low:Day1    -0.22851196 -0.56072511  0.10370118 0.4338483
## High:Day1-Low:Day1      -0.38286713 -0.72990128 -0.03583298 0.0188840
## Low:Day2-Low:Day1        0.03067467 -0.30599791  0.36734724 0.9999986
## Medium:Day2-Low:Day1    -0.21030046 -0.54251361  0.12191268 0.5510054
## High:Day2-Low:Day1      -0.39080122 -0.73783537 -0.04376707 0.0149971
## Low:Day3-Low:Day1        0.01835237 -0.31832021  0.35502495 1.0000000
## Medium:Day3-Low:Day1    -0.16277524 -0.49498838  0.16943790 0.8336101
## High:Day3-Low:Day1      -0.39728617 -0.74432032 -0.05025202 0.0123751
## High:Day1-Medium:Day1   -0.15435517 -0.49706474  0.18835440 0.8894277
## Low:Day2-Medium:Day1     0.25918663 -0.07302651  0.59139977 0.2624020
## Medium:Day2-Medium:Day1  0.01821150 -0.30948153  0.34590452 1.0000000
## High:Day2-Medium:Day1   -0.16228926 -0.50499883  0.18042031 0.8583557
## Low:Day3-Medium:Day1     0.24686433 -0.08534881  0.57907747 0.3259251
## Medium:Day3-Medium:Day1  0.06573672 -0.26195630  0.39342975 0.9993987
## High:Day3-Medium:Day1   -0.16877421 -0.51148378  0.17393536 0.8296575
## Low:Day2-High:Day1       0.41354180  0.06650765  0.76057595 0.0075338
## Medium:Day2-High:Day1    0.17256667 -0.17014290  0.51527623 0.8115666
## High:Day2-High:Day1     -0.00793409 -0.36502928  0.34916111 1.0000000
## Low:Day3-High:Day1       0.40121950  0.05418535  0.74825365 0.0109956
## Medium:Day3-High:Day1    0.22009189 -0.12261768  0.56280146 0.5310462
## High:Day3-High:Day1     -0.01441904 -0.37151423  0.34267616 1.0000000
## Medium:Day2-Low:Day2    -0.24097513 -0.57318827  0.09123801 0.3589979
## High:Day2-Low:Day2      -0.42147589 -0.76851004 -0.07444174 0.0058699
## Low:Day3-Low:Day2       -0.01232230 -0.34899488  0.32435028 1.0000000
## Medium:Day3-Low:Day2    -0.19344991 -0.52566305  0.13876324 0.6603203
## High:Day3-Low:Day2      -0.42796084 -0.77499499 -0.08092669 0.0047701
## High:Day2-Medium:Day2   -0.18050076 -0.52321032  0.16220881 0.7708430
## Low:Day3-Medium:Day2     0.22865283 -0.10356031  0.56086598 0.4329709
## Medium:Day3-Medium:Day2  0.04752522 -0.28016780  0.37521825 0.9999474
## High:Day3-Medium:Day2   -0.18698571 -0.52969527  0.15572386 0.7349843
## Low:Day3-High:Day2       0.40915359  0.06211944  0.75618774 0.0086312
```

```
## Medium:Day3-High:Day2    0.22802598 -0.11468359  0.57073555 0.4811588
## High:Day3-High:Day2     -0.00648495 -0.36358015  0.35061025 1.0000000
## Medium:Day3-Low:Day3    -0.18112761 -0.51334075  0.15108553 0.7357534
## High:Day3-Low:Day3      -0.41563854 -0.76267269 -0.06860439 0.0070562
## High:Day3-Medium:Day3   -0.23451093 -0.57722050  0.10819864 0.4412437
```

4. Test of independence

1. Perform a Pearson's Chi-squared test to check if the Calcium depend on pHtype level?

```
ca.ph.tab <- with(joined.df, table(Calcium, pHtype))
ca.ph.tab
```

```
##         pHtype
## Calcium  acidity natural alkalinity
##   Low         16       1          1
##   Medium      12       1          6
##   High         3       1         12
```

```
chisq.test(ca.ph.tab)
```

```
## Warning in chisq.test(ca.ph.tab): Chi-squared approximation may be
## incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  ca.ph.tab
## X-squared = 18.796, df = 4, p-value = 0.000862
```

2. Why there is a warning from the 4.1? What is an altertive test which we can perform?

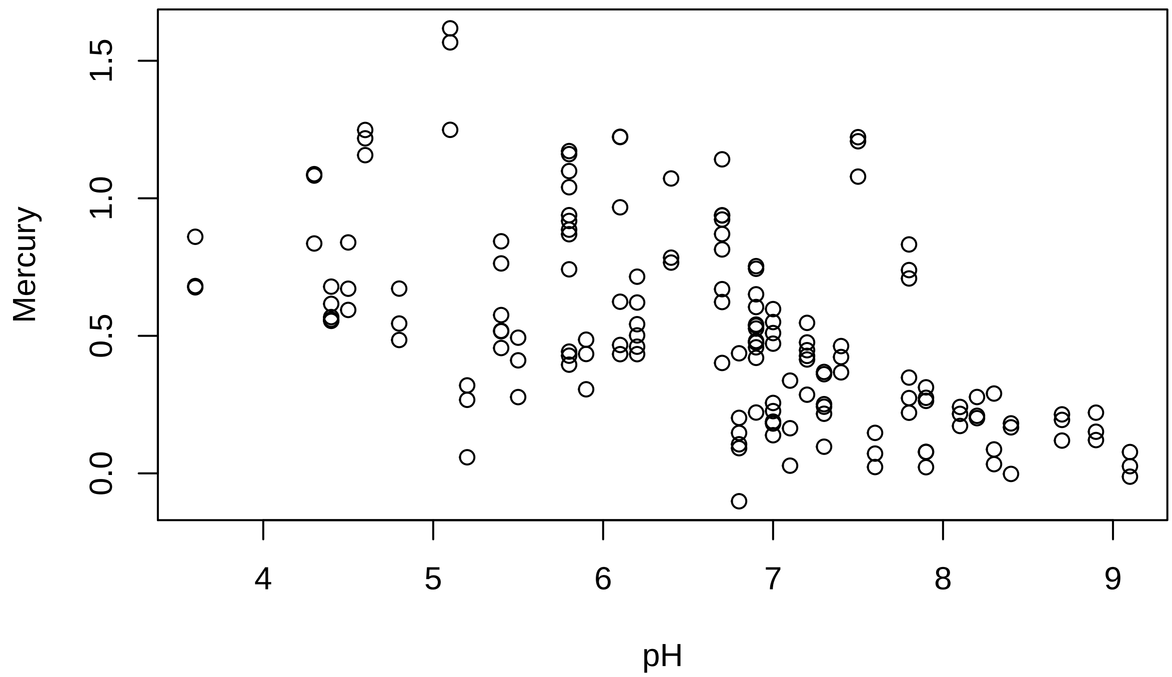
```
fisher.test(ca.ph.tab)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  ca.ph.tab
## p-value = 0.0001327
## alternative hypothesis: two.sided
```

5. Linear regression

1. Produce a scatterplot between pH (x-axis) and mercury (y-axis).

```
with(joined.long.df, plot(pH, Mercury))
```



2. Describe the relationship between pH and mercury showing in the scatterplot.
3. Fit a linear model between pH and mercury, in which mercury is the dependent (response) variable and pH is the independent (explanatory) variable. Name the linear model as `mylm`.

```
mylm <- with(joined.long.df, lm(Mercury ~ pH))
```

4. Use `summary()` to obtain the estimates and test the statistical signifcance of the intercept and slope.

```
summary(mylm)
```

```
##
## Call:
## lm(formula = Mercury ~ pH)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.67553 -0.20926 -0.05343  0.13545  0.86857
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.52662    0.12326  12.385  < 2e-16 ***
## pH          -0.15242    0.01836  -8.301 4.46e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2955 on 157 degrees of freedom
## Multiple R-squared:  0.305,  Adjusted R-squared:  0.3006
```

```
## F-statistic: 68.91 on 1 and 157 DF,  p-value: 4.46e-14
```

5 Write down the equation of the fitted line.

6. Use the following code to check the residuals, for homogeneity of variance.

```
plot(predict(mylm), residuals(mylm))
abline(h = 0, lwd = 2, col = 2)
```

```r
plot(predict(mylm), residuals(mylm))
abline(h = 0, lwd = 2, col = 2)
```