Tutorial 9

ST2137-2420

Material

This tutorial covers the topics and concepts from chapter 8. Think of this topic as a generalisation of the approaches in chapter 7. In chapter 9, we will proceed to linear regression. Take note that 2-sample models, ANOVA models and linear regression models are all linear models.

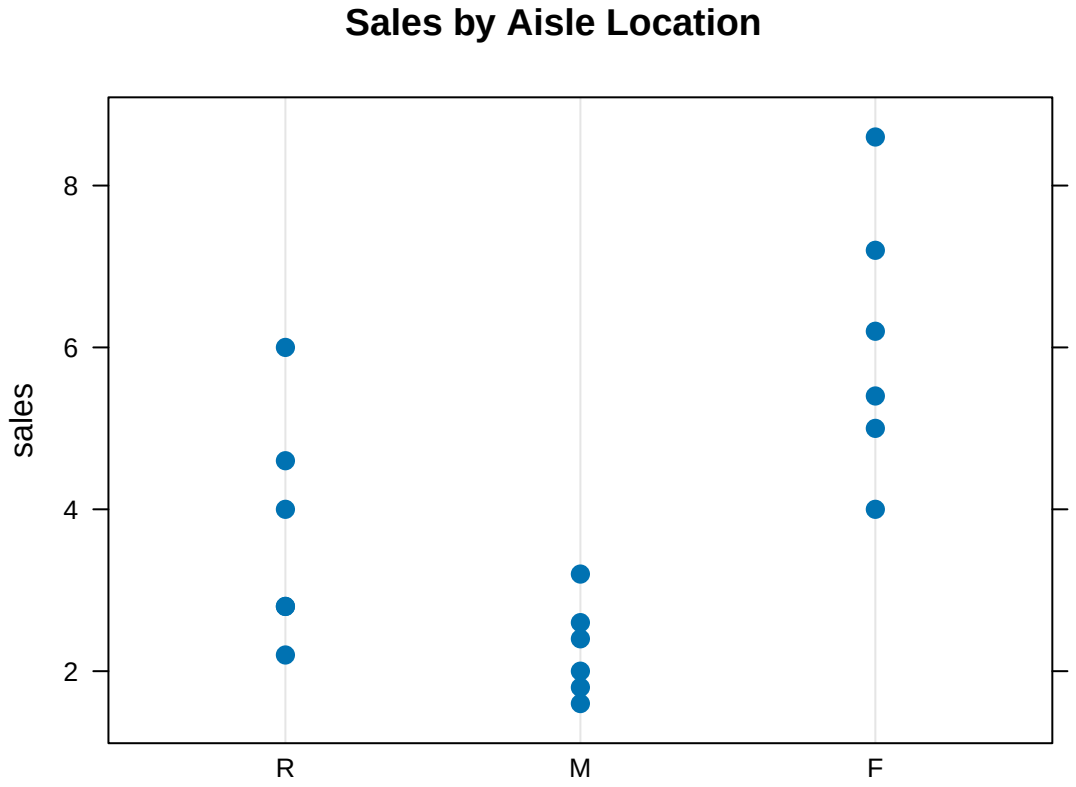
Question 1

The retailing manager of a supermarket chain wants to determine whether product location has any effect on the sale of pet toys. Three different aisle locations are considered: front, middle, and rear. A random sample of 18 stores is selected with 6 stores randomly assigned to each aisle location. The size of the display area and price of the products are constant for all stores. At the end of a one-month trial period, the sales volumes (in thousands of dollars) of the product in each store were recorded in the file `locate.txt`.

1. Assuming that the observations are Normally distributed, use SAS to assess if there is any evidence of a significant difference in average sales among the various aisle locations, at 5% significance level.
2. Boxplots are typically used to assess the distribution within each group. However when we have so few observations, it is sometimes useful to plot every single point, by group. Use `dotplot` from the lattice package in R to create the following plot:

```r
library(lattice)
locate_df <- read.table("data/locate.txt", header=TRUE)

locate_df$location <- factor(locate_df$location, levels=c("R", "M", "F"))
dotplot(sales ~ location, data = locate_df, cex=1.2,
        main="Sales by Aisle Location")
```



4. In R and Python, set the reference level to be "rear". Compute the confidence interval for the differences between (i) front and rear, and (ii) middle and rear. Use a Bonferroni correction to adjust for the multiple tests so that overall, the error rate is 5%.

Solution

The SAS output indicates strong evidence against the null hypothesis. We would reject the null hypothesis and conclude the means are different.

Dependent Variable: sales

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	44.07111111	22.03555556	13.03	0.0005
Error	15	25.36000000	1.69066667		
Corrected Total	17	69.43111111			

In order to compare against the reference "rear", we have to set the levels properly in R. In Python, I recoded the levels so that "rear" is alphabetically first.

R code

```
anova_mod <- lm(sales ~ location, data=locate_df)
confint(anova_mod, level = 1-0.05/2)
```

```
                1.25 %   98.75 %
(Intercept)  2.4116368 5.0550298
locationM   -3.3358278 0.4024944
locationF    0.4641722 4.2024944
```

Python code

```python
import pandas as pd
import numpy as np
from scipy import stats
import statsmodels.api as sm
from statsmodels.formula.api import ols
import statsmodels.stats.multicomp as mc

locate_df = pd.read_table("data/locate.txt", delimiter="\\s+")
locate_df.replace({'F': '3-F', 'M': '2-M', 'R': '1-R'}, inplace=True)

locate_lm = ols('sales ~ C(location, Treatment)', data=locate_df).fit()
#anova_tab = sm.stats.anova_lm(locate_lm, type=3,)
#print(anova_tab)

print(locate_lm.summary(alpha=0.05/2))
```

```
C:\Users\stavg\penvs\p312\Lib\site-packages\scipy\stats\_axis_nan_policy.py:418: UserWarning: `kurtosis
  return hypotest_fun_in(*args, **kwds)
                            OLS Regression Results
==============================================================================
Dep. Variable:                  sales   R-squared:                       0.635
Model:                            OLS   Adj. R-squared:                  0.586
Method:                 Least Squares   F-statistic:                     13.03
Date:                Fri, 11 Apr 2025   Prob (F-statistic):           0.000524
Time:                        13:29:45   Log-Likelihood:                -28.626
No. Observations:                  18   AIC:                             63.25
Df Residuals:                      15   BIC:                             65.92
Df Model:                           2
Covariance Type:            nonrobust
================================================================================================
                                   coef    std err          t      P>|t|      [0.0125      0.9875]
------------------------------------------------------------------------------------------------
Intercept                        3.7333      0.531      7.033      0.000       2.412       5.055
C(location, Treatment)[T.2-M]   -1.4667      0.751     -1.954      0.070      -3.336       0.402
C(location, Treatment)[T.3-F]    2.3333      0.751      3.108      0.007       0.464       4.202
==============================================================================
Omnibus:                        1.188   Durbin-Watson:                   1.186
Prob(Omnibus):                  0.552   Jarque-Bera (JB):                0.789
Skew:                           0.495   Prob(JB):                        0.674
Kurtosis:                       2.732   Cond. No.                         3.73
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Instead of using `confint()`, we can also use the formula in Section 8.4 of the textbook. We will arrive at the same answer:

```r
c1 <- c(-1, 1, 0)
n_vals <- c(6,6,6)
est_coef <- coef(anova_mod)
L <- sum(c1*c(0, est_coef[2:3]))

summary_out <- anova(anova_mod)
MSW <- summary_out$`Mean Sq`[2]
df <- summary_out$Df[2]
se1 <- sqrt(MSW * sum( c1^2 / n_vals ) )
```

```r
q1 <- qt(0.0125, df, 0, lower.tail = FALSE)

lower_ci <- L - q1*se1
upper_ci <- L + q1*se1
cat("The 95% CI for the diff. between the two groups is (",
    format(lower_ci, digits = 6), ", ", format(upper_ci, digits = 6), ").", sep="")
```

```
The 95% CI for the diff. between the two groups is (-3.33583, 0.402494).
```

Question 2

In earlier topics we noticed that, in the student performance dataset from `student-mat.csv`, `G3` scores seem to be different for different `Medu` groups. Remove the group corresponding to `Medu=0` since there are so few observations. Use the following rule to remove outliers from *each group*: X_i is declared an outlier if

$$\frac{|X_i - \text{median}(X)|}{MAD(X)/0.6745} > 2.24$$

Perform the appropriate statistical test(s) to assess the following questions of interest:

5. Is there a significant difference between the 4 groups, at 5% significance level?
6. Estimate the confidence interval for a contrast comparing higher education to non-higher education (i.e. Medu = 4 vs. Medu = 1|2|3).
7. Use Tukey's HSD method to identify which pairs of groups are significantly different from one another at 5% family-wise error level.
8. Repeat the Tukey procedure with all outliers reinstated. How do the results differ?

Solution

R code

```r
stud_perf <- read.table("data/student/student-mat.csv", sep=";",
                        header=TRUE)
stud_perf2 <- stud_perf[stud_perf$Medu != 0, ]
stud_perf2$Medu <- as.factor(stud_perf2$Medu)

remove_outliers <- function(d1) {
  ids <- which(abs(d1 - median(d1))/mad(d1) > 2.24)
  d1[-ids]
}
tmp_list <- vector(mode="list", 4)
for(ii in 1:4){
  tmp_list[[ii]] <- remove_outliers(stud_perf$G3[stud_perf$Medu == ii])
}

lens <- vapply(tmp_list, length, 2L)
stud_perf3 <- data.frame(G3 = unlist(tmp_list), Medu = as.factor(rep(1:4, times=lens)))

lm_outliers_rm <- lm(G3 ~ Medu, data=stud_perf3)
anova(lm_outliers_rm)
```

```
Analysis of Variance Table

Response: G3
           Df Sum Sq Mean Sq F value    Pr(>F)
Medu        3  181.8  60.593  6.2057 0.0004073 ***
```

```
Residuals 348 3397.9   9.764
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value is 0.0004073.

Python code

```
stud_perf  = pd.read_csv("../data/student/student-mat.csv", delimiter=";")
stud_perf2 = stud_perf[stud_perf.Medu != 0]

def identify_outliers(d1):
    id_vec = np.where(.6745 * np.absolute(d1 -
                                          np.quantile(d1, 0.5))/
                                          stats.median_abs_deviation(d1) > 2.24)
    return id_vec

out_df_list = []

for i,df in stud_perf2.groupby('Medu'):
    to_rm = identify_outliers(df.G3)
    out_df = df.drop(df.index[to_rm])
    out_df_list.append(out_df)
stud_perf3 = pd.concat(out_df_list)

lm_outliers_rm = ols('G3 ~ C(Medu, Treatment)', data=stud_perf3).fit()
anova_tab = sm.stats.anova_lm(lm_outliers_rm, type=3,)
print(anova_tab)
```

```
                       df       sum_sq    mean_sq         F    PR(>F)
C(Medu, Treatment)    3.0   181.779541  60.593180  6.205657  0.000407
Residual            348.0  3397.936368   9.764185       NaN       NaN
```

To estimate the contrast, we have to include the coefficients for all levels; I was mistaken in stating that we can leave it out the one for the reference level. Below, I have added the contrast coefficient, group size, and group effect estimate for the reference level (compared to the earlier version of the solution).

R code

```
## higher education vs secondary and below education
c1 <- c(-1/3, -1/3, -1/3, 1)
n_vals <- c(50, 87, 90, 125)
est_coef <- coef(lm_outliers_rm)
L <- sum(c1*c(0, est_coef[2:4]))

summary_out <- anova(lm_outliers_rm)
MSW <- summary_out$`Mean Sq`[2]
df <- summary_out$Df[2]
se1 <- sqrt(MSW * sum( c1^2 / n_vals ) )

q1 <- qt(0.025, df, 0, lower.tail = FALSE)

lower_ci <- L - q1*se1
upper_ci <- L + q1*se1
cat("The 95% CI for the diff. between the two groups is (",
    format(lower_ci, digits = 3), ", ", format(upper_ci, digits = 3), ").", sep="")
```

```
The 95% CI for the diff. between the two groups is (0.746, 2.13).
```

Python code

```
c1 = np.array([-1/3, -1/3, -1/3, 1])
n_vals = np.array([50, 87, 90, 125])

est_params = np.append([0], lm_outliers_rm.params.to_numpy()[1:])
L = np.sum(c1 * est_params)

MSW = lm_outliers_rm.mse_resid
df = lm_outliers_rm.df_resid
q1 = -stats.t.ppf(0.025, df)
se1 = np.sqrt(MSW*np.sum(c1**2 / n_vals))

lower_ci = L - q1*se1
upper_ci = L + q1*se1
print(f"""The 95% CI for the diff. between the two groups is ({lower_ci:.3f}, {upper_ci:.3f}).""")
```

```
The 95% CI for the diff. between the two groups is (0.746, 2.133).
```

SAS Code

```
proc glm data=ST2137.STUD_PERF3;
    class Medu;
    model G3=Medu / clparm;
    means Medu / hovtest=levene welch plots=none;
    lsmeans Medu / adjust=tukey pdiff alpha=.05;
    estimate 'lower_vs_higher' Medu -1 -1 -1 3 / divisor=3;
    run;
quit;
```

The GLM Procedure

Dependent Variable: G3

Parameter	Estimate	Standard Error	t Value	Pr > |t|	95% Confidence Limits	
lower_vs_higher	1.43957088	0.35261373	4.08	<.0001	0.74604871	2.13309305

We can see that R, Python and SAS (you will have to upload the data first) all return the same result. If we perform the estimation using the sum contrast, we should obtain exactly the same value. Here is how we can do so:

```
## Using sum contrasts:
stud_perf3$Medu2 <- stud_perf3$Medu
contrasts(stud_perf3$Medu2) <- contr.sum(4)

lm_outliers_rm_sum <- lm(G3 ~ Medu2, data=stud_perf3)
#anova(lm_outliers_rm_sum)

alpha4 <- -sum(coef(lm_outliers_rm_sum)[-1])
c1 <- c(-1/3, -1/3, -1/3, 1)
n_vals <- c(50, 87, 90, 125)
est_coef <- coef(lm_outliers_rm_sum)
L <- sum(c1*c(est_coef[2:4], alpha4))

summary_out <- anova(lm_outliers_rm)
MSW <- summary_out$`Mean Sq`[2]
df <- summary_out$Df[2]
```

```r
se1 <- sqrt(MSW * sum( c1^2 / n_vals ) )

q1 <- qt(0.025, df, 0, lower.tail = FALSE)

lower_ci <- L - q1*se1
upper_ci <- L + q1*se1
cat("The 95% CI for the diff. between the two groups is (",
    format(lower_ci, digits = 3), ", ", format(upper_ci, digits = 3), ").", sep="")
```

```
The 95% CI for the diff. between the two groups is (0.746, 2.13).
```

The next portion pertains to the multiple comparison tests.

R code

```r
tukey_out <- TukeyHSD(aov(lm_outliers_rm), ordered=TRUE)
tukey_out
```

```
  Tukey multiple comparisons of means
    95% family-wise confidence level
    factor levels have been ordered

Fit: aov(formula = lm_outliers_rm)

$Medu
         diff        lwr      upr     p adj
2-1 0.8519540 -0.5795864 2.283494 0.4168866
3-1 1.0933333 -0.3294728 2.516139 0.1961010
4-1 2.0880000  0.7382076 3.437792 0.0004602
3-2 0.2413793 -0.9714333 1.454192 0.9557730
4-2 1.2360460  0.1097792 2.362313 0.0250490
4-3 0.9946667 -0.1204773 2.109811 0.0993909
```

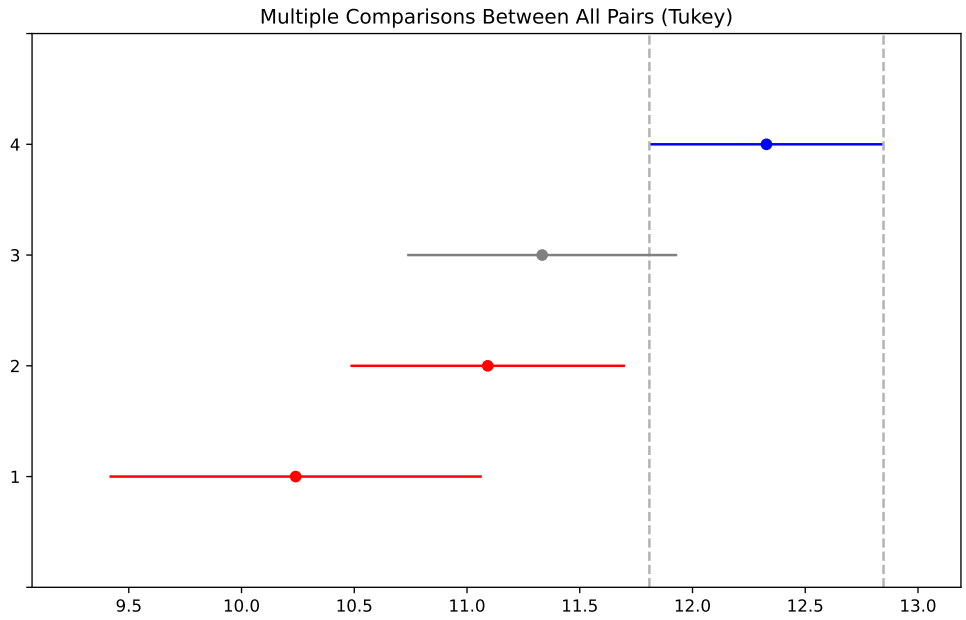
Python code

```python
import statsmodels.stats.multicomp as mc

cp = mc.MultiComparison(stud_perf3.G3, stud_perf3.Medu)
tk = cp.tukeyhsd()
print(tk)
```

```
Multiple Comparison of Means - Tukey HSD, FWER=0.05
====================================================
group1 group2 meandiff p-adj   lower  upper  reject
----------------------------------------------------
     1      2    0.852 0.4169 -0.5796 2.2835  False
     1      3   1.0933 0.1961 -0.3295 2.5161  False
     1      4    2.088 0.0005  0.7382 3.4378   True
     2      3   0.2414 0.9558 -0.9714 1.4542  False
     2      4    1.236  0.025  0.1098 2.3623   True
     3      4   0.9947 0.0994 -0.1205 2.1098  False
----------------------------------------------------
```

```python
tk.plot_simultaneous(comparison_name = 4);
```



The only significant differences are between group 4 and group 1, and between group 4 and group 2.

If we were to apply the procedure on the original data, without `Medu = 0`:

```
lm_outliers <- lm(G3 ~ Medu, data=stud_perf2)
tukey_out2 <- TukeyHSD(aov(lm_outliers), ordered=TRUE)
tukey_out2
```

```
  Tukey multiple comparisons of means
    95% family-wise confidence level
    factor levels have been ordered

Fit: aov(formula = lm_outliers)

$Medu
        diff        lwr      upr     p adj
2-1 1.050189 -0.8337925 2.934171 0.4762184
3-1 1.625064 -0.2727284 3.522857 0.1224640
4-1 3.085393  1.2762245 4.894561 0.0000823
3-2 0.574875 -1.0491895 2.198939 0.7977607
4-2 2.035203  0.5156448 3.554762 0.0033999
4-3 1.460328 -0.0763198 2.996977 0.0693173
```

We can see that, in terms of decisions (significant/not), the outcome does not change whether we keep the outliers in or out. However, notice that in some comparisons the p-value goes up and in some it goes down. In general, please try to make this procedure a habit - consider the analysis with and without any outliers to see how significant they were.