SAS Lab 2 For Statistics 514

Topics:
Chapter 18. ANOVA Diagnostics and Remedial Measures

```
*18.1 Cotton strengths, residual diagnostics;
data cotton;
        do cottontype = 1 to 5;
                do case = 1 to 6;
        input strength @;
                                output;
                end;
        end;
datalines;
7  7 15 11  9 10
12 17 12 18 18 16
14 18 18 19 19 17
19 25 22 19 23 24
 7 10 11 15 11 14
;
proc glm data=cotton;
        title '18.1 anova, cotton';
        class cottontype;
        model strength = cottontype;
        means cottontype;
        output out=cottonout p=pred
                r=resid student=stresid rstudent=stobsresid;
run;
proc print data=cottonout;
        title '18.1 residuals, cotton';
        var cottontype case pred resid;
run;
proc gplot data=cottonout;
        title '18.1 residuals, cotton';
        plot resid*cottontype resid*pred;
run;
proc gplot data=cottonout;
        title '18.1 studentized residual, cotton';
        plot stresid*cottontype stresid*pred;
run;
proc gplot data=cottonout;
        title '18.1 studentized deleted residuals, cotton';
        plot stobsresid*cottontype stobsresid*pred;
run;
proc capability data=cottonout noprint graphics;
        title '18.1 normal probability plot for residuals, cotton';
        probplot resid;
run;
*18.1 Number of pest-free apples;
data apple;
        do pesticide = 1 to 5;
                do case = 1 to 10;
        input pestfree @;
                                output;
                end;
        end;
datalines;
12 13 15 15 16 16 16 17 18 22
28 30 31 32 33 33 37 37 40 41
42 48 48 54 54 55 56 60 63 64
70 71 72 73 75 75 77 79 79 80
92 94 95 96 96 97 97 98 98 99
;
proc glm data=apple;
        title '18.1 anova, apples';
        class pesticide;
        model pestfree = pesticide;
        means pesticide;
        output out=appleout p=pred
                r=resid student=stresid rstudent=stobsresid;
run;
proc print data=appleout;
        title '18.1 residuals, apples';
        var pesticide case pred resid;
run;
proc gplot data=appleout;
        title '18.1 residual plot, apples';
        plot resid*pesticide resid*pred;
run;
proc gplot data=appleout;
        title '18.1 studentized residuals, apples';
        plot stresid*pesticide stresid*pred;
run;
proc gplot data=appleout;
        title '18.1 studentized deleted residual, apples';
        plot stobsresid*pesticide stobsresid*pred;
run;
proc capability data=appleout noprint graphics;
        title '18.1 normal probability plot for residuals, apples';
        probplot resid;
run;
quit;
```

```
*18.2 Cotton strengths, hartley, levene;
data cotton;
        do cottontype = 1 to 5;
                do case = 1 to 6;
        input strength @;
                                output;
                end;
        end;
datalines;
7   7 15 11  9 10
12 17 12 18 18 16
14 18 18 19 19 17
19 25 22 19 23 24
 7 10 11 15 11 14
;
proc anova data=cotton;
        title '18.2 modified levene test, variance constant?';
        class cottontype;
        model strength = cottontype;
        means cottontype / hovtest = levene (type=abs);
run;
*18.2 Number of pest-free apples;
data apple;
        do pesticide = 1 to 5;
                do case = 1 to 10;
        input pestfree @;
                                output;
                end;
        end;
datalines;
12 13 15 15 16 16 16 17 18 22
28 30 31 32 33 33 37 37 40 41
42 48 48 54 54 55 56 60 63 64
70 71 72 73 75 75 77 79 79 80
92 94 95 96 96 97 97 98 98 99
;
proc anova data=apple;
        title '18.2 modified levene test, apple';
        class pesticide;
        model pestfree = pesticide;
        means pesticide / hovtest = levene (type=abs);
run;
quit;
```

```
*Weighted least squares, unequal variance;
data soil;
        do soiltype = 1 to 3;
                do case = 1 to 5;
        input y @;
                        if soiltype = 1 then x1 = 1;
                        if soiltype = 1 then x2 = 0;
                        if soiltype = 1 then x3 = 0;
                        if soiltype = 2 then x1 = 0;
                        if soiltype = 2 then x2 = 1;
                        if soiltype = 2 then x3 = 0;
                        if soiltype = 3 then x1 = 0;
                        if soiltype = 3 then x2 = 0;
                        if soiltype = 3 then x3 = 1;
                        x = 1;
                        output;
                end;
        end;
datalines;
365.04 372.41 368.71 361.41 357.81
249.64 247.15 244.69 239.85 242.26
149.90 148.41 146.94 145.47 151.41
;
PROC reg DATA = SOIL;
        title 'UNweighted LS method, FULL regression';
        MODEL y = X1 X2 x3 / noint;
        output out=soilout p=pred r=resid;
RUN;
PROC reg DATA = SOILout;
        title 'UNweighted LS method, REDUCED regression';
        MODEL y = X / NOINT;
RUN;
proc glm data=soil;
        title 'UNweighted ANOVA method';
        class soiltype;
        model y = soiltype;
run;
proc print data=soilboth;
        title 'UNweighted output';
        var soiltype x1 x2 x3 y pred resid;
run;
proc means data=soil noprint;
        var y;
        by soiltype;
        output out=varout std=sd;
run;
proc sort data=varout;
        by soiltype;
run;
proc sort data=soil;
        by soiltype;
run;
data soilboth;
        merge soilout varout;
                by soiltype;
run;
proc gplot data=soilboth;
        title '18.3 residuals, soil';
        plot resid*pred;
run;
DATA SOILboth;
        * weights used in weighted regression calculated;
        SET SOILboth;
        WT = 1/(sd*sd);
RUN;
PROC reg DATA = SOILboth;
        title 'weighted least squares, FULL regression';
        WEIGHT WT;
        MODEL y = X1 X2 x3 / noint;
        output out=soil2 p=pred2 r=resid2;
RUN;
PROC REG DATA = SOILboth;
        title 'weighted least squares, REDUCED regression';
        MODEL y = X / NOINT;
        WEIGHT WT;
RUN;
proc print data=soil2;
        title 'weighted output';
        var soiltype x1 x2 x3 sd wt y pred2 resid2;
run;
QUIT;
```

```
*Transforming, box-cox for unequal variance;
data soil;
        prod = 1;
        do type = 1 to 3;
                do case = 1 to 5;
        input y @;
                        *least squares form of data;
                        if type = 1 then x1 = 1;
                        if type = 1 then x2 = 0;
                        if type = 1 then x3 = 0;
                        if type = 2 then x1 = 0;
                        if type = 2 then x2 = 1;
                        if type = 2 then x3 = 0;
                        if type = 3 then x1 = 0;
                        if type = 3 then x2 = 0;
                        if type = 3 then x3 = 1;
                        x = 1;
                        output;
                end;
        end;
datalines;
365.04 372.41 368.71 361.41 357.81
249.64 247.15 244.69 239.85 242.26
149.90 148.41 146.94 145.47 151.41
;
PROC reg DATA = SOIL;
        title 'UNweighted LS method';
        MODEL y = X1 X2 x3 / noint;
        output out=soilout p=pred r=resid;
RUN;
proc gplot data=soilout;
        title '18.4 residuals, soil';
        plot resid*pred;
run;
proc means data=soil noprint;
        title '18.4 means and variances, soil';
        var y;
        by type;
        output out=meansout std=sd mean=mn;
run;
data varsoil;
        set meansout;
        varprpmean = sd*sd/mn;
        varprpmean2 = sd*sd/(mn**2);
        varprpmean4 = sd*sd/(mn**4);
run;
proc print data=varsoil;
        title '18.4 variance/mean ratios';
        var mn sd varprpmean varprpmean2 varprpmean4;
run;
data product;
        *product for boxcox transformations;
        *product is a scalar;
        set soil end = last;
        retain temp;
        if _n_ = 1 then temp = y;
        mult = temp*y;
        if last = 1 then output;
        temp = mult;
run;
proc print data=product;
        title '18.4 box-cox transformed data';
        var mult;
run;
data box;
        *boxcox transformations;
        if _n_ = 1 then set product;
        set soil;
        k2 = mult**(1/15);
        lambda = 2.6;
        *array of 15 rows x 5 columns;
        array transf[5] t1-t5;
        do i = 1 to 5;
                lambda = lambda + 0.1;
                transf[i] = (1/(lambda*k2**(lambda-1)))*(y**lambda - 1);
        end;
run;
proc print data=box;
        title '18.4 box-cox transformed data';
        var mult k2 y t1 t2 t3 t4 t5;
run;
proc anova data=box;
        title '18.4 box-cox, lambda 2.7';
        class type;
        model t1 = type;
run;
proc anova data=box;
        title '18.4 box-cox, lambda 2.8';
        class type;
        model t2 = type;
run;
proc anova data=box;
        title '18.4 box-cox, lambda 2.9';
        class type;
        model t3 = type;
run;
proc anova data=box;
        title '18.4 box-cox, lambda 3.0';
        class type;
        model t4 = type;
run;
proc anova data=box;
        title '18.4 box-cox, lambda 3.1';
        class type;
        model t5 = type;
run;
data suggested;
        *using suggested transformations;
        set box;
        array suggest[4] s1-s4;
        *suggested trans from residual plot;
        s1 = sqrt(y);
        s2 = sqrt(y + 1);
        *suggested trans from table;
        s3 = log(y);
        *suggested trans from box-cox;
        s4 = y**2.9;
run;
proc anova data=suggested;
        title '18.4 tranformation y = sqrt(y)';
        class type;
        model s1 = type;
run;
proc anova data=suggested;
        title '18.4 tranformation y = sqrt(y + 1)';
        class type;
        model s2 = type;
run;
proc anova data=suggested;
        title '18.4 tranformation y = ln(y)';
        class type;
        model s3 = type;
run;
proc anova data=suggested;
        title '18.4 tranformation y = y^2.9';
        class type;
        model s4 = type;
run;
QUIT;
```

```
*Winding speeds, residual diagnostics,p 787;
data winding;
        input breaks speed case;
datalines;
4       1       1
3       1       2
2       1       3
3       1       4
4       1       5
4       1       6
5       1       7
6       1       8
5       1       9
4       1       10
2       1       11
4       1       12
4       1       13
2       1       14
3       1       15
4       1       16
7       2       1
6       2       2
4       2       3
6       2       4
7       2       5
2       2       6
9       2       7
5       2       8
5       2       9
9       2       10
8       2       11
4       2       12
6       2       13
7       2       14
6       2       15
6       2       16
12      3       1
6       3       2
14      3       3
12      3       4
10      3       5
9       3       6
12      3       7
17      3       8
7       3       9
6       3       10
12      3       11
11      3       12
6       3       13
13      3       14
10      3       15
14      3       16
17      4       1
15      4       2
7       4       3
20      4       4
11      4       5
11      4       6
16      4       7
25      4       8
11      4       9
24      4       10
18      4       11
21      4       12
16      4       13
19      4       14
9       4       15
23      4       16
;
proc glm data=winding;
        title '18.17(a) means,anova';
        class speed;
        model breaks = speed;
        means speed;
        output out=windingout p=pred r=resid;
run;
proc print data=windingout;
        title '18.17(a) residuals';
        var speed case pred resid;
run;
proc gplot data=windingout;
        title '18.17(b) residual plot';
        plot resid*speed;
run;
proc anova data=windingout;
        title '18.17(c) modified levene test, variance constant?';
        class speed;
        model breaks = speed;
        means speed / hovtest = levene (type=abs);
run;
*3.14(B) COX-BOX TRANSFORMATIONS;
DATA winding;
        *arrays are matrices!;
        ARRAY OBS1(64) OBS101-OBS164;
        ARRAY OBS2(64) OBS201-OBS264;
        ARRAY OBS3(64) OBS301-OBS364;
        ARRAY T1(64) T101-T164;
        ARRAY T2(64) T201-T264;
        ARRAY T3(64) T301-T364;
        ARRAY T4(64) T401-T464;
        ARRAY T5(64) T501-T564;
        DO i = 1 TO 64;
                INPUT OBS1[i] OBS2[i] obs3[i];
        END;
        PROD = OBS1[1];
        DO i = 2 TO 63;
                PROD = PROD*OBS1[i+1];
        END;
        K2 = PROD**(1/64);
        DO i = 1 TO 64;
                K1 = 1/(-0.2*K2**(-0.2-1));
                T1[i] = K1*(OBS1[i]**(-0.2)-1);
                K1 = 1/(-0.1*K2**(-0.1-1));
                T2[i] = K1*(OBS1[i]**(-0.1)-1);
                T3[i] = K2*LOG(OBS1[i]);
                K1 = 1/(0.1*K2**(0.1-1));
                T4[i] = K1*(OBS1[i]**(0.1)-1);
                K1 = 1/(0.2*K2**(0.2-1));
                T5[i] = K1*(OBS1[i]**(0.2)-1);
        END;
DATALINES;
4       1       1
3       1       2
2       1       3
3       1       4
4       1       5
4       1       6
5       1       7
6       1       8
5       1       9
4       1       10
2       1       11
4       1       12
4       1       13
2       1       14
3       1       15
4       1       16
7       2       1
6       2       2
4       2       3
6       2       4
7       2       5
2       2       6
9       2       7
5       2       8
5       2       9
9       2       10
8       2       11
4       2       12
6       2       13
7       2       14
6       2       15
6       2       16
12      3       1
6       3       2
14      3       3
12      3       4
10      3       5
9       3       6
12      3       7
17      3       8
7       3       9
6       3       10
12      3       11
11      3       12
6       3       13
13      3       14
10      3       15
14      3       16
17      4       1
15      4       2
7       4       3
20      4       4
11      4       5
11      4       6
16      4       7
25      4       8
11      4       9
24      4       10
18      4       11
21      4       12
16      4       13
19      4       14
9       4       15
23      4       16
;
DATA windingbox;
        SET winding;
        ARRAY OBS1(64) OBS101-OBS164;
        ARRAY OBS2(64) OBS201-OBS264;
        ARRAY OBS3(64) OBS301-OBS364;
        ARRAY T1(64) T101-T164;
        ARRAY T2(64) T201-T264;
        ARRAY T3(64) T301-T364;
        ARRAY T4(64) T401-T464;
        ARRAY T5(64) T501-T564;
        DO i = 1 TO 64;
                windb1 = T1[i];
                windb2 = T2[i];
                windb3 = T3[i];
                windb4 = T4[i];
                windb5 = T5[i];
                speed = obs2[i];
                case = obs3[i];
                OUTPUT;
        END;
        KEEP windb1 windb2 windb3 windb4 windb5 speed case;
RUN;
proc print data=windingbox;
        var windb1 windb2 windb3 windb4 windb5 speed case;
run;
proc anova data=windingbox;
        title '18.17(e) box-cox, lambda -0.2';
        class speed;
        model windb1 = speed;
run;
proc anova data=windingbox;
        title '18.17(e) box-cox, lambda -0.2';
        class speed;
        model windb2 = speed;
run;
proc anova data=windingbox;
        title '18.17(e) box-cox, lambda -0.2';
        class speed;
        model windb3 = speed;
run;
proc anova data=windingbox;
        title '18.17(e) box-cox, lambda -0.2';
        class speed;
        model windb4 = speed;
run;
proc anova data=windingbox;
        title '18.17(e) box-cox, lambda -0.2';
        class speed;
        model windb5 = speed;
run;
quit;
```

```
*18.18 WINDING SPEEDS, ANOVA DIAGNOSIS;
*AND REMEDIAL MEASURES, PAGE 788;
DATA WINDING;
        INPUT BREAKS SPEED CASE;
        LOGBREAKs = LOG10(BREAKS);
DATALINES;
4       1       1
3       1       2
2       1       3
3       1       4
4       1       5
4       1       6
3       1       7
6       1       8
5       1       9
4       1       10
2       1       11
4       1       12
4       1       13
2       1       14
3       1       15
4       1       16
7       2       1
6       2       2
4       2       3
6       2       4
7       2       5
2       2       6
9       2       7
5       2       8
5       2       9
9       2       10
3       2       11
8       2       12
6       2       13
4       2       14
7       2       15
6       2       16
12      3       1
6       3       2
14      3       3
12      3       4
10      3       5
9       3       6
12      3       7
17      3       8
7       3       9
6       3       10
12      3       11
11      3       12
6       3       13
13      3       14
10      3       15
14      3       16
17      4       1
15      4       2
7       4       3
20      4       4
13      4       5
11      4       6
16      4       7
25      4       8
11      4       9
24      4       10
18      4       11
21      4       12
16      4       13
19      4       14
9       4       15
23      4       16
;
PROC GLM DATA=WINDING;
        TITLE '18.18(A) LOG(10) WINDING SPEEDS, ANOVA, PAGE 788';
        CLASS SPEED;
        MODEL LOGBREAKs = SPEED;
        MEANS SPEED;
        OUTPUT OUT=WINDINGout PREDICTED=BREAKP RESIDUALS=RESID;
RUN;
PROC PRINT DATA=WINDINGout;
        TITLE '18.18(A) LOG(10) WINDING SPEEDS, RESIDUALS, PAGE 788';
        VAR SPEED CASE BREAKP RESID;
RUN;
*18.18(B) WINDING SPEEDS, RESIDUAL PLOT, PAGE 788;
PROC GPLOT DATA=WINDINGout;
        TITLE '18.18(B) LOG(10) WINDING SPEEDS, RESIDUAL PLOT, PAGE 788';
        PLOT RESID*SPEED;
RUN;
proc capability data=windingout noprint graphics;
        title '18.18(b) normal probability plot for residuals';
        probplot resid;
run;
PROC SORT DATA=windingout;
        BY RESID;
RUN;
PROC REG DATA=windingout OUTEST=EST NOPRINT;
        MODEL resid = breaks;
        OUTPUT OUT=OUTPLOT PREDICTED=PRED RESIDUAL=RESID;
RUN;
DATA windingout;
        SET OUTPLOT NOBS=NOBS;
        QUANTILE = PROBIT( (_N_- (3/8)) / (NOBS + (1/4)) );
RUN;
DATA OUTPLOT2;
        IF _N_ = 1 THEN SET EST;
        SET windingout;
        EXPRESIDUAL = _RMSE_*QUANTILE;
RUN;
PROC GPLOT DATA=OUTPLOT2;
        TITLE '18.18(b) NORMAL PROBABILITY PLOT';
        PLOT RESID*EXPRESIDUAL;
RUN;
PROC CORR DATA=outplot2;
        TITLE '18.18(b) correlation of normal prob plot';
  VAR RESID EXPRESIDUAL;
RUN;
proc anova data=windingout;
        title '18.18(c) modified levene test, variance constant?';
        class speed;
        model logbreaks = speed;
        means speed / hovtest = levene (type=abs);
run;
QUIT;
```