

# STAT 580 Homework 3

Yifan Zhu

February 27, 2017

- 1.
- 2.
- 3.

```
#include <stdio.h>

#define N 16 /* number of observations */
#define P 2 /* number of predictors */

void dgels_(char *TRANS, int *m, int *n, int *NRHS, double *A, int *LDA,
            double *B, int *LDB, double *WORK, int *LWORK, int *INFO);

int main()
{
    /* longley dataset from R: Employed (Y) GNP.deflator and Population
       (X) */
    double Y[N] = {60.323, 61.122, 60.171, 61.187, 63.221, 63.639,
                    64.989,
                    63.761, 66.019, 67.857, 68.169, 66.513, 68.655,
                    69.564,
                    69.331, 70.551
    };
    double X[N][P] =
    {
        {83, 107.608},
        {88.5, 108.632},
        {88.2, 109.773},
        {89.5, 110.929},
        {96.2, 112.075},
        {98.1, 113.27},
        {99, 115.094},
        {100, 116.219},
        {101.2, 117.388},
        {104.6, 118.734},
        {108.4, 120.445},
        {110.8, 121.95},
        {112.6, 123.366},
        {114.2, 125.368},
        {115.7, 127.852},
        {116.9, 130.081}
    };
};
```

```

    char trans = 'N';
    int m = N;
    int n = P+1;
    int nrhs = 1;
    int lwork = 2 * m*n;
    double work[lwork];
    int info;
    int i, j;

    double A[m * n];
    double B[m];

    for (i=0; i<m; i++){
A[i]=1;
}

    for (i = 0; i < m; i++)
    {
        for (j = 1; j < n+1; j++)
        {
            A[j * m + i] = X[i][j-1];
        }
    }

    for (i = 0; i < m; i++)
    {
        B[i] = Y[i];
    }

    dgels_(&trans, &m, &n, &nrhs, A, &m, B, &m, work, &lwork, &info);

    if (info != 0)
    {
        printf("dgels_error_%d\n", info);
    }
    else
    {
        printf("The_regression_coefficients:_");
        for (i = 0; i < n; i++)
        {
            printf("%.6f\t", B[i]);
        }
        printf("\n");
    }
return 0;
}

```

4.

```

#include <stdio.h>
#include <stdlib.h>
#include <assert.h>

#define N 16 /* number of observations */
#define P 2 /* number of predictors */

```

```

void dgesvd_(char *JOBV, char *JOBVT, int *m, int *n, double *A, int *
    LDA, double *S, double *U, int *LDU, double *VT, int *LDVT, double *
    WORK, int *LWORK, int *INFO);

int main()
{
    char jobv = 'S';
    char jobvt = 'A';
    int m = N;
    int n = P;
    double A[m * n];
    double s[n];
    double u[m * n];
    double vt[n * n];
    double *work;
    int lwork = -1;
    double lworkopt;
    int i, j, info;

    /* longley dataset from R */
    double X[N][P] =
    {
        {83, 107.608},
        {88.5, 108.632},
        {88.2, 109.773},
        {89.5, 110.929},
        {96.2, 112.075},
        {98.1, 113.27},
        {99, 115.094},
        {100, 116.219},
        {101.2, 117.388},
        {104.6, 118.734},
        {108.4, 120.445},
        {110.8, 121.95},
        {112.6, 123.366},
        {114.2, 125.368},
        {115.7, 127.852},
        {116.9, 130.081}
    };

    double Xbar[P];
    for (j = 0; j < P; j++)
    {
        Xbar[j] = 0;
        for (i = 0; i < N; i++)
        {
            Xbar[j] = Xbar[j] + X[i][j];
        }
        Xbar[j] = Xbar[j] / (double) N ;
    }

    for (i = 0; i < m; i++)
    {
        for (j = 0; j < n; j++)
        {
            A[j * m + i] = X[i][j] - Xbar[j];

```

```

    }
}

dgesvd_(&jobu, &jobvt, &m, &n, A, &m, s, u, &m, vt, &n, &lworkopt, &
    lwork, &info);

if (info != 0)
{
    printf("The_dgesvd_error_%d\n", info);
}
else
{
    lwork = (int) lworkopt;
    work = (double *) malloc(lwork * sizeof(double));
    assert(work != NULL);

    dgesvd_(&jobu, &jobvt, &m, &n, A, &m, s, u, &m, vt, &n, work, &
        lwork, &info);

    if (info != 0)
    {
        printf("The_dgesvd_error_%d\n", info);
    }
    else
    {
        printf("The_principal_component_scores:\n");
        for (i = 0; i < m; i++)
        {
            for (j = 0; j < n; j++)
            {
                printf("%.6f\t", u[j * m + i] * s[j]);
            }
            printf("\n");
        }
    }
}
return 0;
}

```

5.

```

#include <stdio.h>

#define N 10

int main(){
    double x[N] = {3.1, -1.2, 5.3, 1, 4.4, 21, 3, 7, -1.2, 3.2};
    int i, j;
    double temp;
    for (i = 1; i < N; i++){
        j = i;
        while (j > 0 && x[j-1] > x[j]){
            temp = x[j-1];
            x[j-1] = x[j];
            x[j] = temp;
            j--;
        }
    }
}

```

```

    }
    printf("Sorted_data:\n");
    for (i=0; i < N; i++){
        printf("%f_", x[i]);
    }

    printf("\n_Median:\n");
    if (N%2 == 0){
        printf("%f\n", (x[N/2 -1] + x[N/2])/2.0);
    } else {
        printf("%f\n", x[N/2]);
    }
    return 0;
}

```