Variance Analysis for Kernel Smoothing of a Varying-Coefficient Model With Longitudinal Data

Jinsong Chen

A Thesis Submitted to the
University of North Carolina at Wilmington in Partial Fulfillment
Of the Requirements for the Degree of
Master of Arts

Department of Mathematics and Statistics

University of North Carolina at Wilmington

2003

Approved by

Advisory Committee

Chair

Accepted by

Dean, Graduate School

This thesis has been prepared in the style and format

Consistent with the journal

American Mathematical Monthly.

TABLE OF CONTENTS

ABSTRACT

We consider the estimation of the $\mathbf{k}$ + 1-dimensional nonparametric component $\beta(t)$ of the varying-coefficient model $Y(t) = X^T(t)\beta(t) + \varepsilon(t)$ based on longitudinal observation $(Y_{ij}, X_i(t_{ij}), t_{ij}), i = 1, ..., n, j = 1, ..., n_i$, where t_{ij} is the jth observed design time point t of the ith subjects at t_{ij}. The subjects are independently selected, but the repeated measurements within subject are possibly correlated.A Monte Carlo Simulation was established, kernel smoothing method was used to estimate $\beta(t)$ that minimizes a local least square criterion. The distribution for $\varepsilon(t)$ was analyzed. The degree of freedom was investigated.

DEDICATION

This thesis is dedicated to my parents in China.

ACKNOWLEDGMENTS

I would like to espress my gratitude to my thesis advisor, Dr. James Blum, whose estensice knowledge, dedication and enthusiasm made this project achievable. I wish to thank all committee members, Dr. Dargan Frierson and Dr. Susan Simmons. Their advises to complete this thesis is much appreciated. I also would like to thank Dr. Russell Herman for his help on LaTex issues.

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

In a longitudinal study, outcomes and covariates are observed from different subjects each repeatedly measured at a set of distinct time points. This type of data is common in medical and epidemiological studies. Let $t_{ij}, j = 1, ..., n_i$ be the times over which the measurements of the i^{th} subject took place. Let Y_{ij} be the observed response and $X_i(t_{ij})$ be the observed real-valued outcome and covariates for the i_{th} subject at time t_{ij}. The measurements, $(Y_{ij}, X_i(t_{ij}), t_{ij}), i = 1, ...n, j = 1, ..., n_i$, are independent between different subjects, but can be correlated at different time points within a subject. We consider here the linear time-varying coefficient model of form.

$$Y_{ij} = X_i^T(t_{ij})\beta(t_{ij}) + \varepsilon_i(t_{ij}) \tag{1}$$

Where $X_i(t) = (1, X_{i1}(t), ..., X_{ik}(t))^T$ and $\varepsilon_i(t)$ are independent, $t_{ij} \in R$, and the $\varepsilon_i(t)$ are mean 0 stochastic processes. $\beta(t) = (\beta_0(t), ..., \beta_k(t))^T$, and $\beta_l(t) \in R$ for all $l = 0, ..., k$. (Wu, Chiang, Hoover, 1998)

In this paper, one application to longitudinal data is presented. The data considered here involve covariates of infants' genders and HIV infection status (HIV positive or negative) measured one year after birth and the third trimester maternal vitamin A levels during pregnancy and repeatedly measured weights of 328 infants from an African AIDS cohort study at the Johns Hopkins University (Hoover, Rice, Wu and Yang, 1996). All infants were born from HIV infected mothers in central Africa and survived beyond one year of age. The research continued two years and infants' weights were recorded during every scheduled monthly visit. Due to various reasons, a number of the scheduled visits were missed by some infants which resulted in unequal numbers of repeated weight measurements per infant. The main objective is to evaluate the time-varying effects of two binary covariates (child's gender and HIV status), and one continuous covariate (the third trimester maternal vitamin A level) on the children's weights. Previous studies have shown that vitamin A can improve immune function and resistance to disease [cf.Semba (1994)]. Biologically, a significant association between maternal vitamin A levels and infant growth may suggest the benefit of vitamin A supplementation in the mother's and infant's diet.

In this application, we use the actual measurements and fit the data to (1) with $X_{i10} = \cdots = X_{in_i0} = 1$,

$$X_{i11} = \cdots = X_{in_i1} = \begin{cases} 1 & \text{if the } i\text{th infant is HIV positive}, \\ 0 & \text{if the } i\text{th infant is HIV negative}, \end{cases}$$

$$X_{i12} = \cdots = X_{in_i2} = \text{the } i\text{th infant's maternal vitamin A level},$$

$$X_{i13} = \cdots = X_{in_i3} = \begin{cases} 1 & \text{if the } i\text{th infant is male}, \\ 0 & \text{if the } i\text{th infant is female}, \end{cases}$$

$$Y_{ij} = \text{weight in kilograms of the } i\text{th infant at time } t_{ij} \text{ after birth},$$

This data set was analyzed by Hoover et al.(1996) using kernel and spline methods. The smoothing results of kernel methods is presented in here. Figure 1 shows the estimated values of $\beta_l(t), l = 0, ..., 3$, together with their ± 2 point-wise bootstrap standard error bands. From the figure it is seen that the magnitudes of the coefficients of all three factors initially increase with time and then level off. The initial increase with time probably reflects the cumulative effects of additional diseases early in life due to HIV infection and/or low vitamin A levels. The leveling off of the difference may be due to the establishment of the infants immunity function at one year of age and frailty effects from the sickest and lowest weight babies dyeing. Besides using bootstrap standard errors to assess variability, there are some other important inferential issues. Various types of confidence regions might be desired: for example, intervals for components or linear combinations of components of $\beta(t)$ for fixed t and simultaneous confidence bands for all t in an interval.

2 ESTIMATION BY KERNEL SMOOTHING METHOD

Theory and applications of estimates based on kernel, spline and locally weighted polynomial methods have been extensively studied in the literature for nonparametric curve estimation with independent cross-sectional data. With properly selected smoothing parameters, these estimation methods have good asymptotic properties such as optimal rates of convergence, and usually give reliable results in real applications. Thus it is natural to extend these methods to the estimation for observations from longitudinal studies, in this paper, kernel smoothing methods are used.

According model (1), if $E(X(t)X^T(t))$ is invertible, the $\beta(t)$ is unique and given by

$$\beta(t) = E(X(t)X^T(t))^{-1}E(X(t)Y(t))$$

Here, we use kernel estimation method. The advantage is its flexibility of form and mathematical tracability. Kernel estimators are linear estimators in the sense that we can express the value of the estimator at any point t as a weighted sum of the responses. The weights in this sum all derive from a kernel function. Define for a general kernel K:

$$K_h(t) = \frac{1}{h}K(\frac{t}{h})$$

The parameter h is called the bandwidth or smoothing parameter. The bandwidth determines how far away observations are allowed to be from t and still contribute to the estimation of $\beta(t)$. The bandwidth also governs the peakedness of the weight function and, hence, the degree of dependence of the estimator on information near t. Small values of h will result in rougher (wigglier) estimators that rely heavily on the data near t. In contrast, larger h's allow more averaging to occur and thereby give smoother estimators. Figure 2 shows how the kernel estimator fits to the data for different bandwidth selections(Eubank, 1999).

We would like our kernel function to satisfy the moment conditions

$$\int_{-\infty}^{\infty} K(u)\, du = 1$$

The above condition is roughly equivalent to having the weights sum to one.

$$\int_{-\infty}^{\infty} uK(u)\, du = 0$$

This is a type of symmetry condition that is automatically satisfied if K is symmetric about zero.

$$M_2 = \int_{-\infty}^{\infty} u^2 K(u)\, du \neq 0$$

and condition

$$V = \int_{-\infty}^{\infty} K(u)^2 < \infty.$$

Here we use Gaussian kernel function as:

$$K_h(t) = \frac{1}{h\sqrt{2\pi}} exp(-\frac{1}{2}(\frac{t}{h})^2)$$

The kernel estimates are developed based on finding the unique $\beta(t) = (\beta_0(t), ..., \beta_k(t))^T$, which minimizes the locally weighted least squares criterion

$$L_M(t) = \sum_{i=1}^{n} \sum_{j=1}^{n_i} [Y_{ij} - \sum_{l=0}^{k} X_{ijl}\beta_l(t)]^2 K(\frac{t - t_{ij}}{h})$$

Where $M = \sum_{i=1}^{n} n_i$ is the total number of observations, h is a positive bandwidth which might depend on M, and $K(.)$ is Borel measurable kernel function mapping R onto R.

Let Y_i and X_i be the outcome vector and design matrix of ith subject: $Y_i = (Y_{i1}, ..., Y_{in_i})^T$

and

$$X_i = \begin{pmatrix} X_{il0} & X_{il1} & \cdots & X_{ilk} \\ \vdots & \vdots & \ddots & \vdots \\ X_{in_i0} & X_{in_i1} & \cdots & X_{in_ik} \end{pmatrix}$$

Let $K_i(t)$ be the diagonal matrix:

$$K_i(t) = diag(K[(t - t_{i1})h^{-1}], ..., K[(t - t_{in_i})h^{-1}])$$

It is convenient to rewrite $L_M(t)$ into the following matrix form

$$L_M(t) = \sum_{i=1}^{n} (Y_i - X_i\beta(t))^T K_i(t)(Y_i - X_i\beta(t))$$

Then the estimate of β is

$$\hat{\beta}(t) = (\sum_{i=1}^{n} X_i^T K_i(t) X_i)^{-1} (\sum_{i=1}^{n} X_i^T K_i(t) Y_i)$$

The estimation of $\hat{\beta}(t)$ depends on the choices of the bandwidth and the kernel function.

Besides the kernel estimate, there are other nonparametric estimates, such as smoothing spline and locally weighted polynomial. Splines are piece-wise polynomial which are joined smoothly at knots. Statistical properties and practical implementation of spline methods can be found in Eubank(1988) among others. Locally weighted polynomials are generalization of the kernel type estimates, for which theory and applications with independent cross-sectional data have been studied by Stone (1977), Cleveland (1979), Buja, Hastie and Tibshriani (1989, Hastie and Tibshriani (1990), Fan (1993) among others. This generalization have many advantages over the kernel methods, particularly in estimation at boundary points.

In this paper, we focus on the estimates of the variance σ^2 of the error terms ε_{ij} in model (1), then obtain an indication of the variability of the probability distributions of

Y. For a standard bivariate linear regression model $Y_i = \beta_o + \beta_1 X_i + \varepsilon_i$, the residual is $\varepsilon_i = Y_i - \hat{Y}_i$, the sum of square is $\sum_{i=1}^{n}(Y_i - \hat{Y})^2$, the sample variance is

$$s^2 = \frac{\sum_{i=1}^{n}(Y_i - \hat{Y})^2}{n-2}$$

which is the residual sum of squares divided by degrees of freedom.

Corresponding to the model (1) in this paper, the residuals are $\varepsilon_{ij} = Y_{ij} - \hat{Y}_{ij}$, here, $\hat{Y}_{ij} = X_{ij}\hat{\beta}(t_{ij})$. Hence the appropriate sum of square is $\sum_{i=1}^{n}\sum_{j=1}^{n_i}(Y_{ij} - \hat{Y}_{ij})^2$. The resulting estimator is an extension of the usual sample variance:

$$s^2 = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n_i}(Y_{ij} - \hat{Y}_{ij})^2}{\sum_{i=1}^{n}(n_i - 1)}$$

We know that the variance σ^2 can be estimated by the sample variance s^2. In here, we want to see how good the observed sample variance s^2 fit the variance σ^2. Also we can explore the distribution of s^2.

3 MONTE CARLO SIMULATION

For simplicity, we consider model(1) with a time-independent covariate $X = (1, X_1, X_2, X_3)^T$, where X_1 and X_2 are two Bernoulli random variables, the probability for happening of 1 or 0 is equally to 0.5. X_3 is a $N(0, 0.25)$ random variable. The coefficient curves are given by

$$\beta_0(t) = 15 + 20\sin(\frac{t\pi}{60})$$

$$\beta_1(t) = 4 - (\frac{t-20}{10})^2$$

$$\beta_2(t) = 2 - 3cos(\frac{(t-25)\pi}{15})^2$$

$$\beta_3(t) = -5 + \frac{(30-t)^3}{5000}$$

These coefficient curves are similar to the estimated curves in longitudinal study. A simple random sample of N subjects $X_i, i = 1, ..., N$ was generated for (X_1, X_2, X_3) based on the joint density.

$$f(X_1, X_2, X_3) = \frac{0.5}{(2\pi)^{1/2}} exp(-2X_3^2) \times 1_{\{0,1\}}(X_1) \times 1_{\{0,1\}}(X_2) \times 1_{\{-\infty,\infty\}}(X_3)$$

To create design time points, we generated 30 equally spaced ”scheduled” time points and i random diaplacement points S_{i1} from the $U(0, 1)$ distribution such that $S_{i1} = S_{i1} + (l - 1), l = 1, ..., 30$, in addition, each ”scheduled” time point S_{il} had a probability of $m\%(m = 0, 20, 40, 60)$ of being randomly missing. The remaining observed time points were denoted by t_{ij}. This led to unequal numbers of repeated measurements n_i and different observed time points t_{ij} per subject. The random errors $\varepsilon_i(t_{ij})$ were generated according to the mean 0 Gaussian process with covariance matrix:

$$COV[\varepsilon_{i_1}(t_{i_1 j_1}), \varepsilon_{i_2}(t_{i_2 j_2})] = \begin{cases} 4exp(-|t_{i_1 j_1} - t_{i_2 j_2}|) & if \quad i_i = i_2 \\ 0 & if \quad i_1 \neq i_2 \end{cases}$$

The outcomes Y_{ij} were obtained by substituting the observed $(t_{ij}, X_i, \varepsilon_i(t_{ij}))$ and the foregoing coefficient curves into model (1). The kernel function used in this simulation is:

$$K(t) = \frac{1}{h\sqrt{2\pi}} e^{\frac{-t^2}{2h^2}}$$

We have different subjects number as $N = 50, 100, 150, 200$, missing data value $m = 0, 20, 40, 60$, kernel bandwidth as $h = 0, 1.0, 1.5, 2.0,$, giving 64 simulations.

We wish to determine the distribution of the estimated variance. It should have mean equal to the true variance σ^2 (which is 4 for these cases), It is hypothesized that, the sample variance of data generated is:

$$s^2 \sim \frac{\sigma^2}{k} \chi_k^2$$

where, s^2 is the variance estimated from the set of simulations. χ_k^2 is a random variable that follows the chi-square distribution with k degrees of freedom, have a mean of k and variance of $2k$.

For each case, PROC UNIVARIATE in SAS is used to construct the mean and variance of the variance estimate, $\overline{x}_{s^2}$ and $\hat{\sigma}_{s^2}^2$ respectively. We know for a random variable x with variance σ_x^2, the random variable ax (a is a constant) has a variance of $a^2\sigma_x^2$. Thus the variance of $\frac{\sigma^2}{k}\chi_k^2$, we have $\hat{\sigma}_{s^2}^2 \approx \frac{2(\sigma^2)^2}{k}$, then k value can be approximated by

$$k \approx \frac{2(\sigma^2)^2}{\hat{\sigma}_{s^2}^2} = \frac{32}{\hat{\sigma}_{s^2}^2}$$

Then we can transform the simulated to the form $\frac{k}{\sigma^2}s^2$ from variance data and create a probability plot to estimate agreement with the Chi-Square distribution, taking the χ^2 as a $Gamma(\alpha, \beta)$ with $\alpha = \frac{k}{2}$ and $\beta = 2$. A probability plot is much like a Q-Q plot (only the horizontal scale differs). Both compare ordered values of a variable with quantiles of a specified theoretical distribution. If the data distribution matches the theoretical distribution, the points on the plot form a linear pattern.

4 CONCLUSION

We expect s^2 has a multiple of a chi-square distribution with k degrees of freedom. In the probability plots, the simulated results match the theoretical Gamma distribution very well. The quantile plot of results fit the line with light tail at the end. When the number of subjects increases with the same bandwidth and missing value, the fit is better. For example, Figure 10 shows the improvement of the fit for the case with bandwidth=1.0 and missing=60%. Chart A for 50 subject is the worst fit, then the fit improves as we progress to chart D with 200 subjects, which has the best fit. So, there is an improvement in the chi-square approximation for larger numbers of subjects.

According to $k = 2\alpha$, here α is the estimated shape parameter for Gamma distribution from probability plot, we can build a table for the k values, this is shown in Table 1. For increasing numbers of subjects or decrease in missing time points, the degrees of freedom increase due to the larger number of observations. For example, for the case with bandwith=0.5 and missing=60%, the k value changes from 548.3 to 1967.4 for an increase of subjects from 50 to 200. Corresponding to changing of bandwidth, there are no obvious trends appearing, but for the case with 200 subjects, the degrees of freedom increase as the bandwidth increases.

Table 2 show the average values for the variance estimates. The value of mean increases for the larger numbers of subjects. For example, for the case with bandwith=1 and missing=60%, the mean increases from 3.9 to 4.3 as the number of subjects increases from 50 to 200. This is most likely due to the fact that a greater number of subjects would allow for a smaller smoothing parameter, while oversmoothing results in a higher estimate of residual error variance.

5 REFERENCES

Cleveland, W.S. (1979). Robust Locally-weighted Regression and Smoothing Scatterplots. *Journal of American Statistic Association*, 74, 829-836, 1979.

Randall L. Eubank, *Nonparametric Regression and Spline Smoothing*, New York, 1999

Fan, J.Q. Local Linear Regression Smoothers and Their Minimax Efficiencies. *Ann.Statist.* 21, 196-216, 1993.

Jianqing Fan, Jin-Ting Zhang. Functional Linear Models for Longitudinal Data, *Journal of the Royal Statistical Society*, B62, 303-322, 2000.

Fuja, A., Hastie, T.J. and Tibshriani, R.J. Linear Smoothers and Additive Models. *Ann. Statist.* 17, 453-555, 1989.

Hastie, T.J. and Tibshriani, R.J. *Generalized Additive Models*. London, 1990.

Donald R. Hoover, John A.Rice, Colin O. Wu, Li-Ping Yang (1996). Nonparametric Smoothing Estimates of Time-Varying Coefficient Models With Longitudinal Data, *Biometrika*, 85, 809-822, 1998.

Stone, C.J. Consistent Nonparametric Regression. *Ann. Statist.* 5, 549-645, 1977

Colin O Wu, Chin-Tsang Chiang, and Donald R. Hoover, Asymptotic Confidence Regions for Kernel Smoothing of a Varying-Coefficient Model With Longitudinal Data. *Journal of the American Statistical Association*, Vol.93,No.444, 1998

6 TABLES AND FIGURES

Table 1: Table of k value

		Subjects			
Bandwith	Missing	50	100	150	200
0.5	60	548.3	1042.9	1632.3	1967.4
	40	758.6	1491.9	2309.4	3003.5
	20	929.0	1865.7	2755.8	3712.9
	0	1206.1	2005.6	3367.2	4544.4
1	60	530.6	1015.4	1586.7	2090.0
	40	726.4	1627.8	2411.2	2948.9
	20	955.5	1942.8	2858.0	3749.8
	0	1125.7	2308.4	3488.0	4206.2
1.5	60	572.0	1121.2	1538.5	2037.8
	40	784.4	1477.4	2277.4	3104.1
	20	1024.2	1855.8	2754.7	3785.7
	0	1201.7	2376.0	3254.7	4508.7
2	60	494.2	1011.9	1576.6	2207.0
	40	753.6	1544.6	2315.6	3288.8
	20	1029.1	1850.8	2609.0	3838.2
	0	1096.6	2122.0	3402.6	4727.4

Table 2: Mean of Sample Variance

		Subjects			
Bandwith	Missing	50	100	150	200
0.5	60	3.6	4.0	4.0	4.1
	40	3.7	4.0	4.0	4.0
	20	3.7	4.0	4.0	4.1
	0	3.8	4.0	4.0	4.1
1	60	3.9	4.1	4.2	4.3
	40	3.9	4.1	4.1	4.2
	20	3.9	4.1	4.1	4.1
	0	3.9	4.0	4.0	4.1
1.5	60	4.1	4.3	4.3	4.3
	40	4.1	4.2	4.2	4.3
	20	4.1	4.2	4.2	4.2
	0	4.1	4.1	4.2	4.2
2	60	4.3	4.5	4.5	4.5
	40	4.3	4.4	4.4	4.4
	20	4.2	4.3	4.3	4.4
	0	4.2	4.3	4.3	4.3

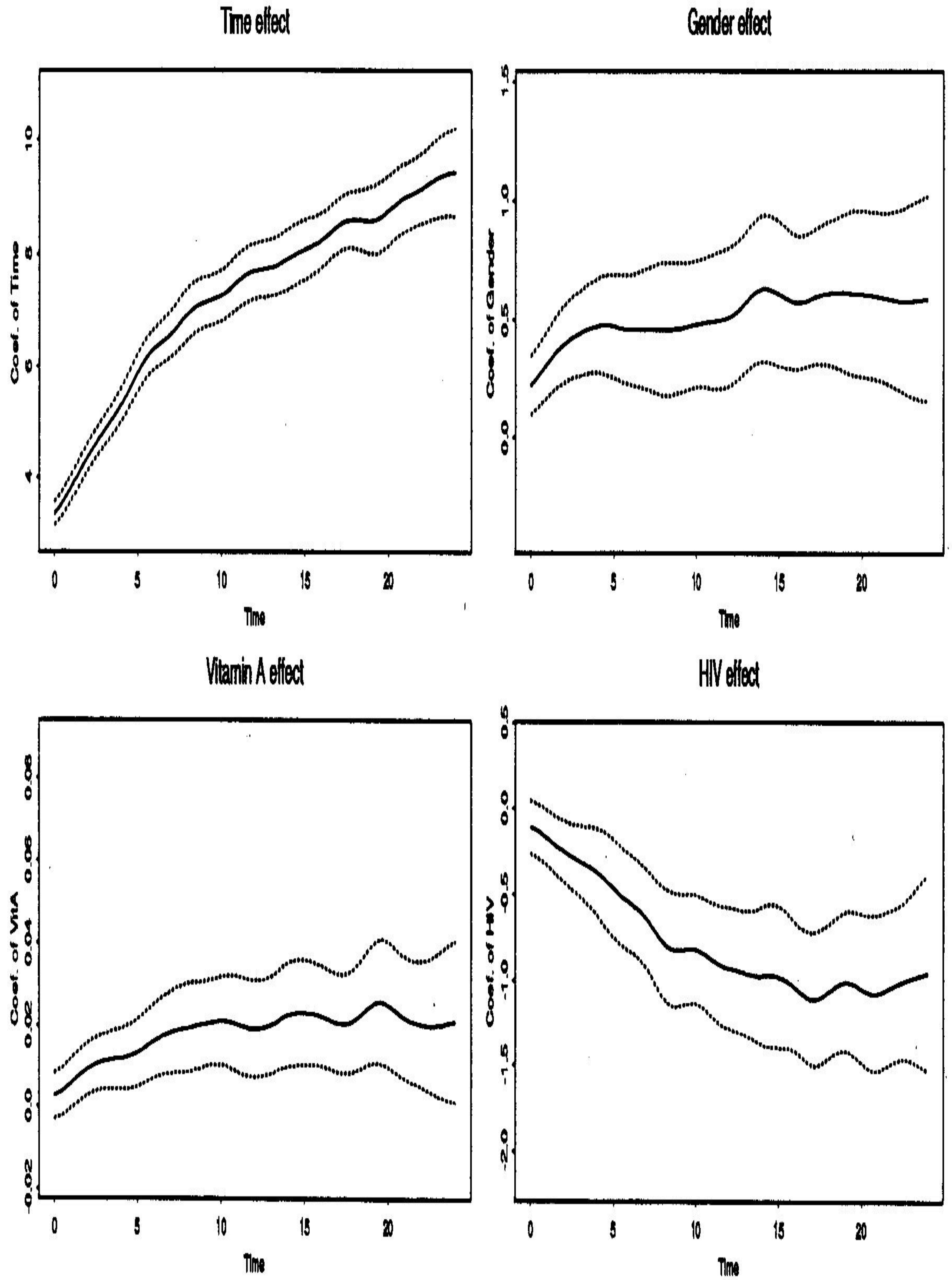

Figure 1: Estimates, predictions and residuals using kernel method with the standard Gaussian kernel and h=1.2 as the bandwidth. The dashed curves represent the ±2 bootstrap standard error bands. Time effect: $/beta_0(t)$ vs. time. HIV effect: $\hat{\beta}_1(t)$ vs. time. Vitamin A effect: the estimated effect of vitamin A $\hat{\beta}_2(t)$ vs. time. Gender effect: $\hat{\beta}_3(t)$ vs. time. (Hoovers, 1998)

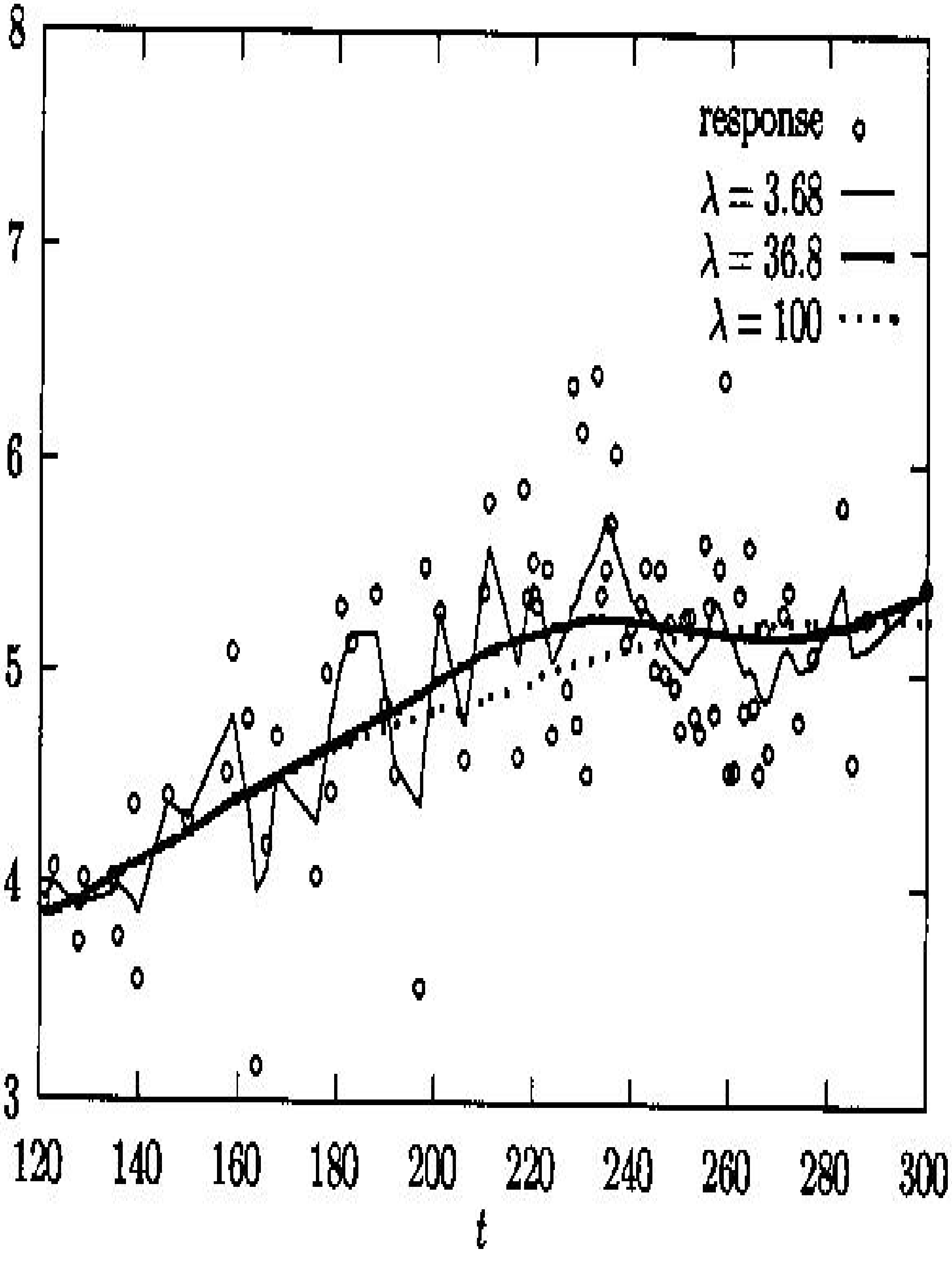

Figure 2: Kernel Fits to Assay Data. (Eubank, 1999)

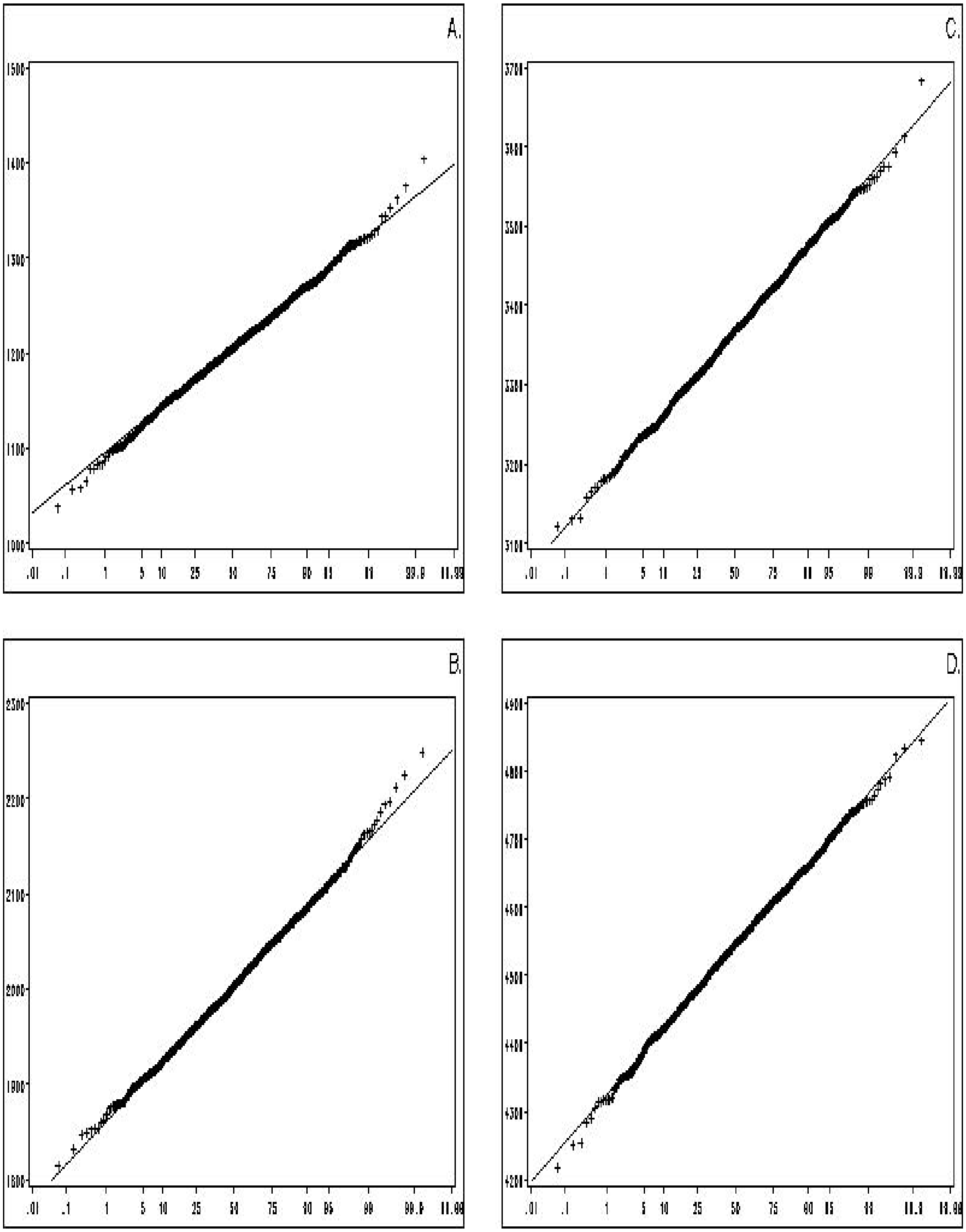

Figure 3: Probability plot for bandwidth=0.5 missing 0%. Horizontal axis represent the theoretical chi-square distribution quantile. Vertical axis represent the observed values. A) 50 subjects B) 100 subjects C) 150 subjects D) 200 subjects

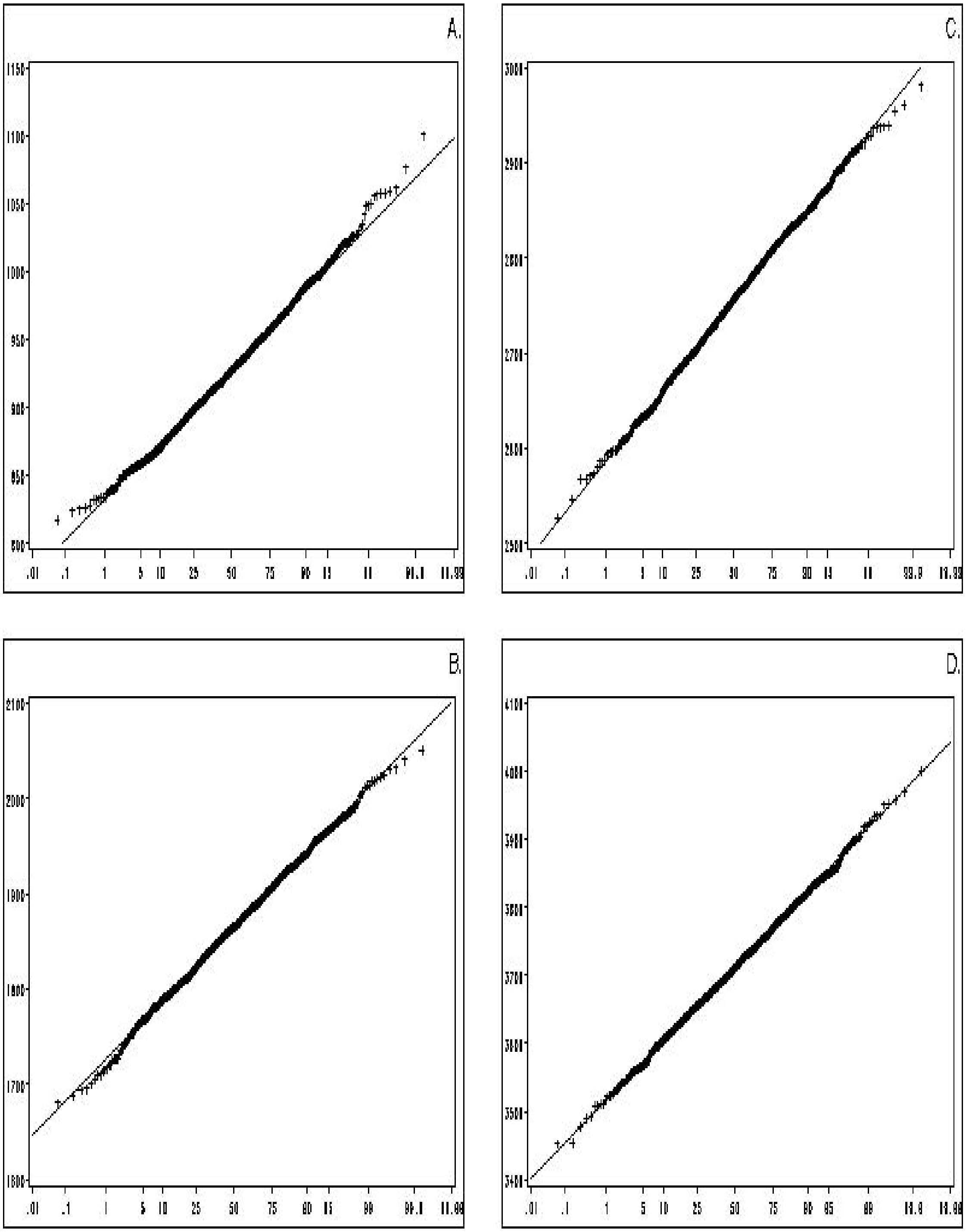

Figure 4: Probability plot for bandwidth=0.5 missing 20%. Horizontal axis represent the theoretical chi-square distribution quantile. Vertical axis represent the observed values. A) 50 subjects B) 100 subjects C) 150 subjects D) 200 subjects

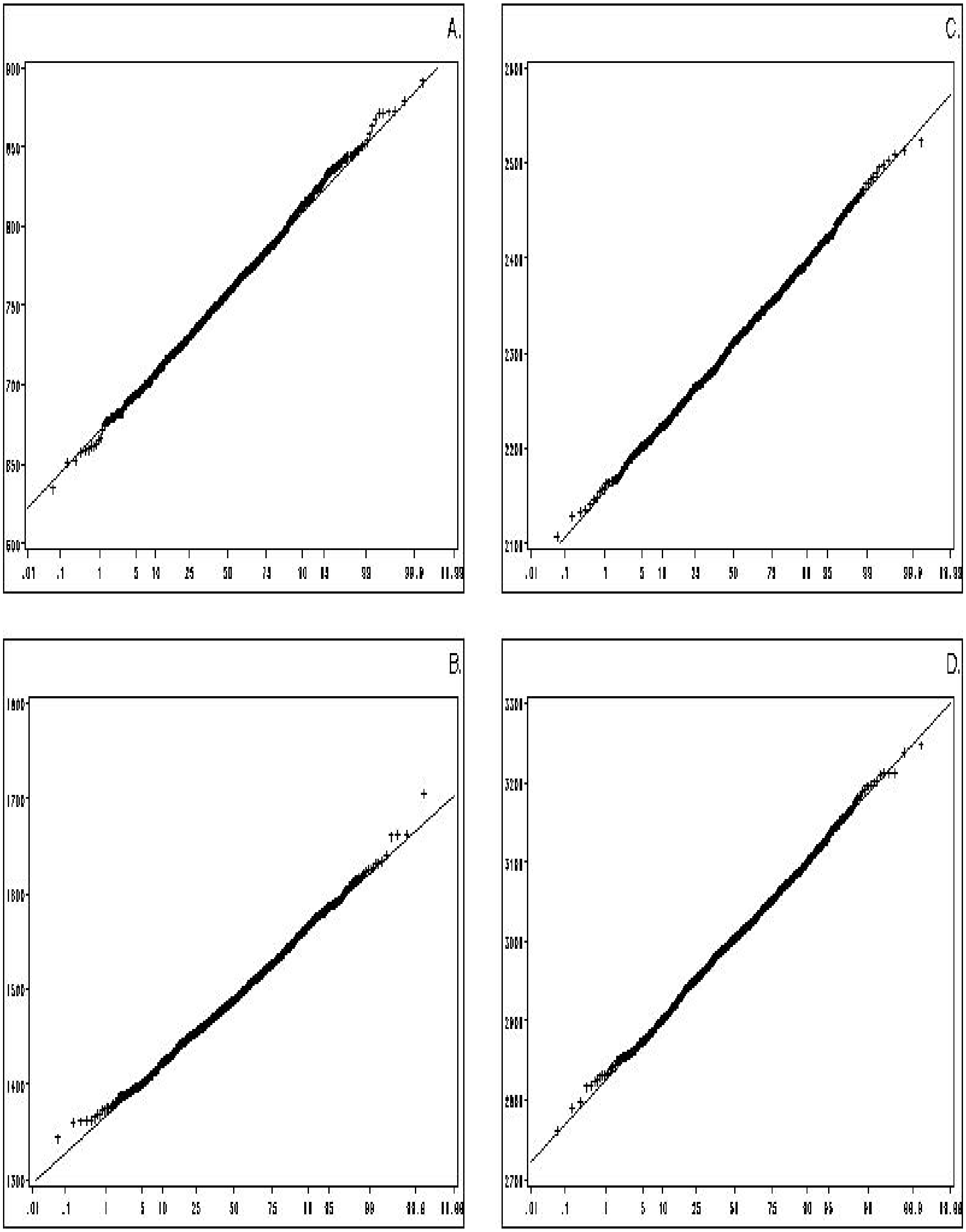

Figure 5: Probability plot for bandwidth=0.5 missing 40%. Horizontal axis represent the theoretical chi-square distribution quantile. Vertical axis represent the observed values. A) 50 subjects B) 100 subjects C) 150 subjects D) 200 subjects

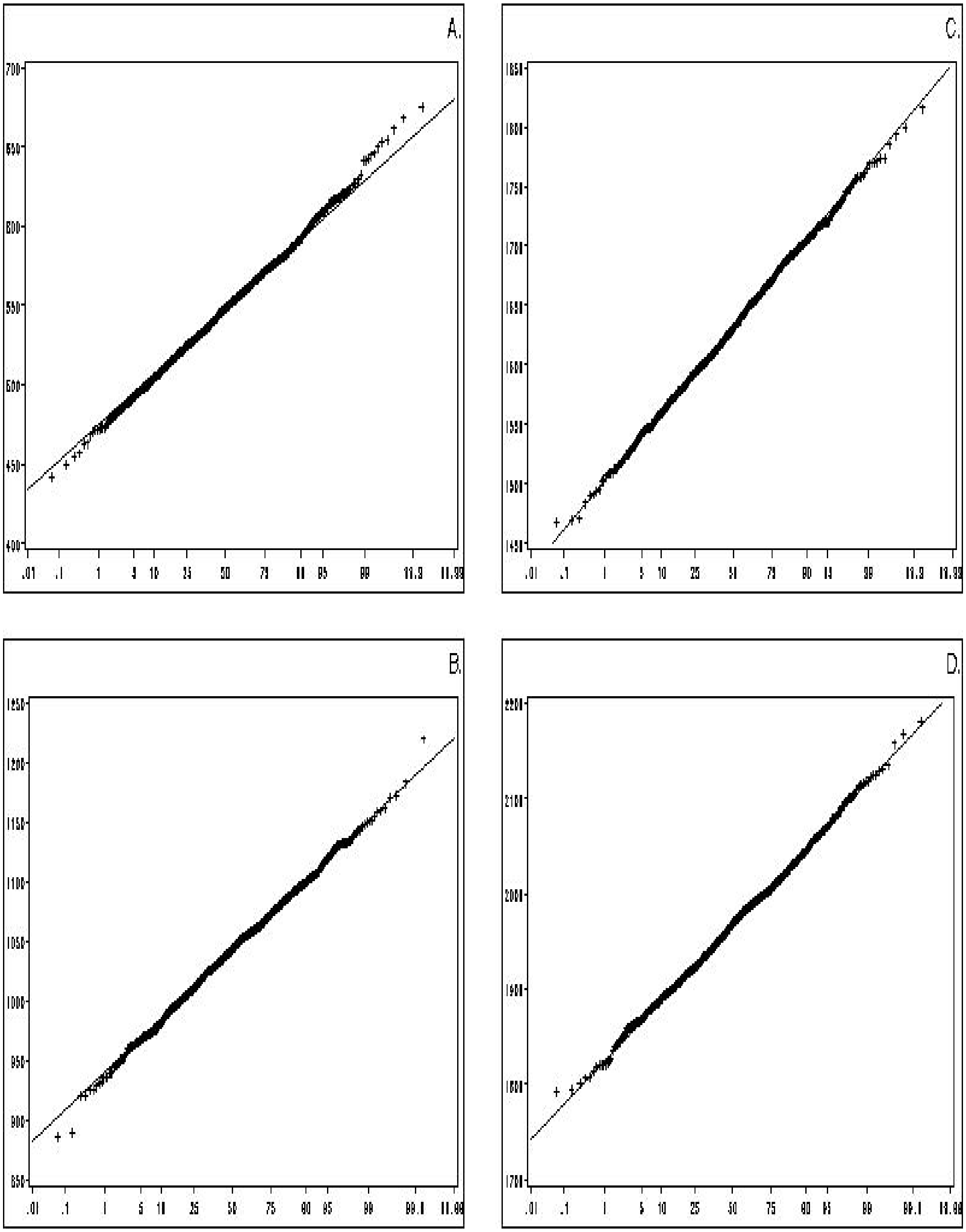

Figure 6: Probability plot for bandwidth=0.5 missing 60%. Horizontal axis represent the theoretical chi-square distribution quantile. Vertical axis represent the observed values. A) 50 subjects B) 100 subjects C) 150 subjects D) 200 subjects

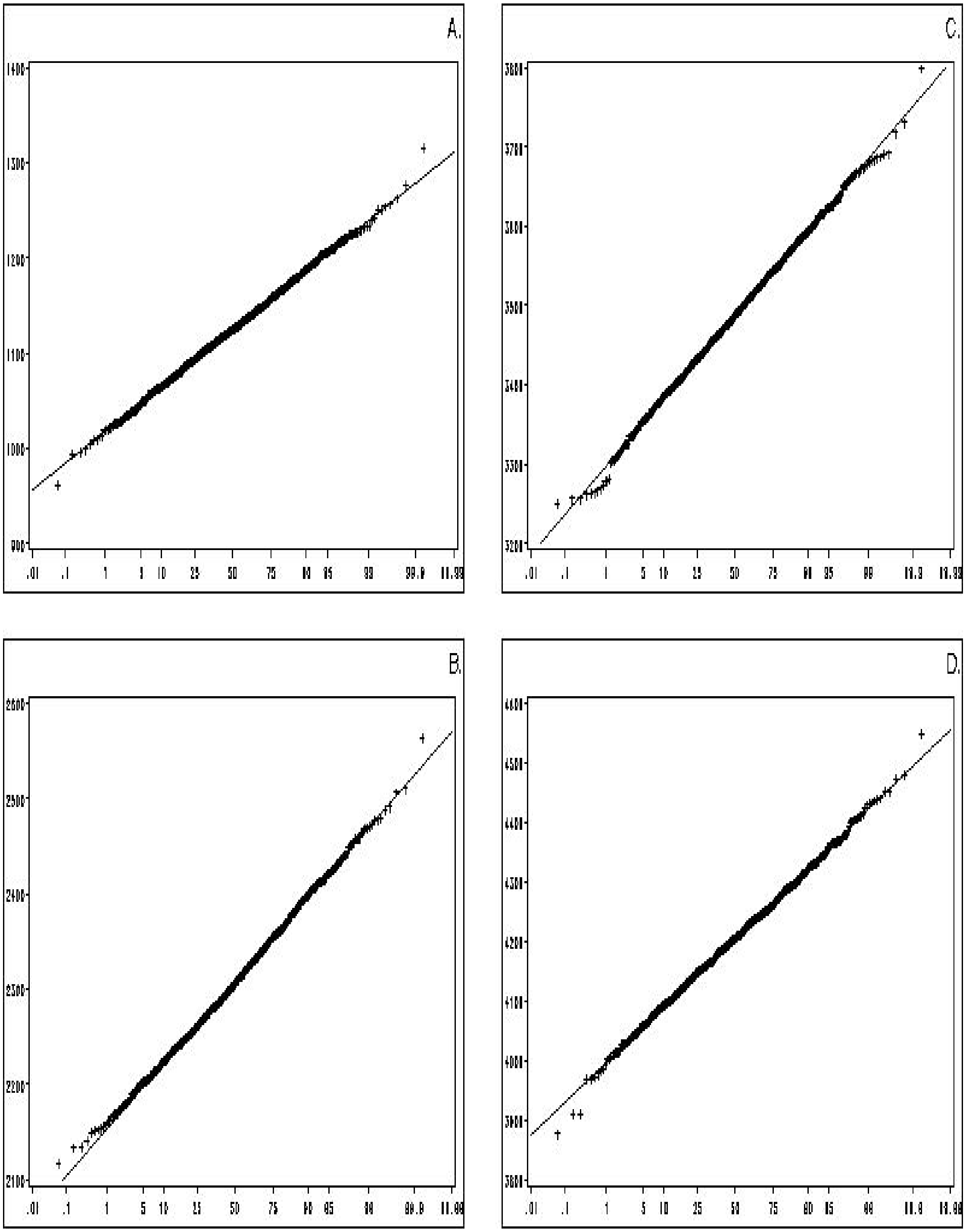

Figure 7: Probability plot for bandwidth=1.0 missing 0%. Horizontal axis represent the theoretical chi-square distribution quantile. Vertical axis represent the observed values. A) 50 subjects B) 100 subjects C) 150 subjects D) 200 subjects

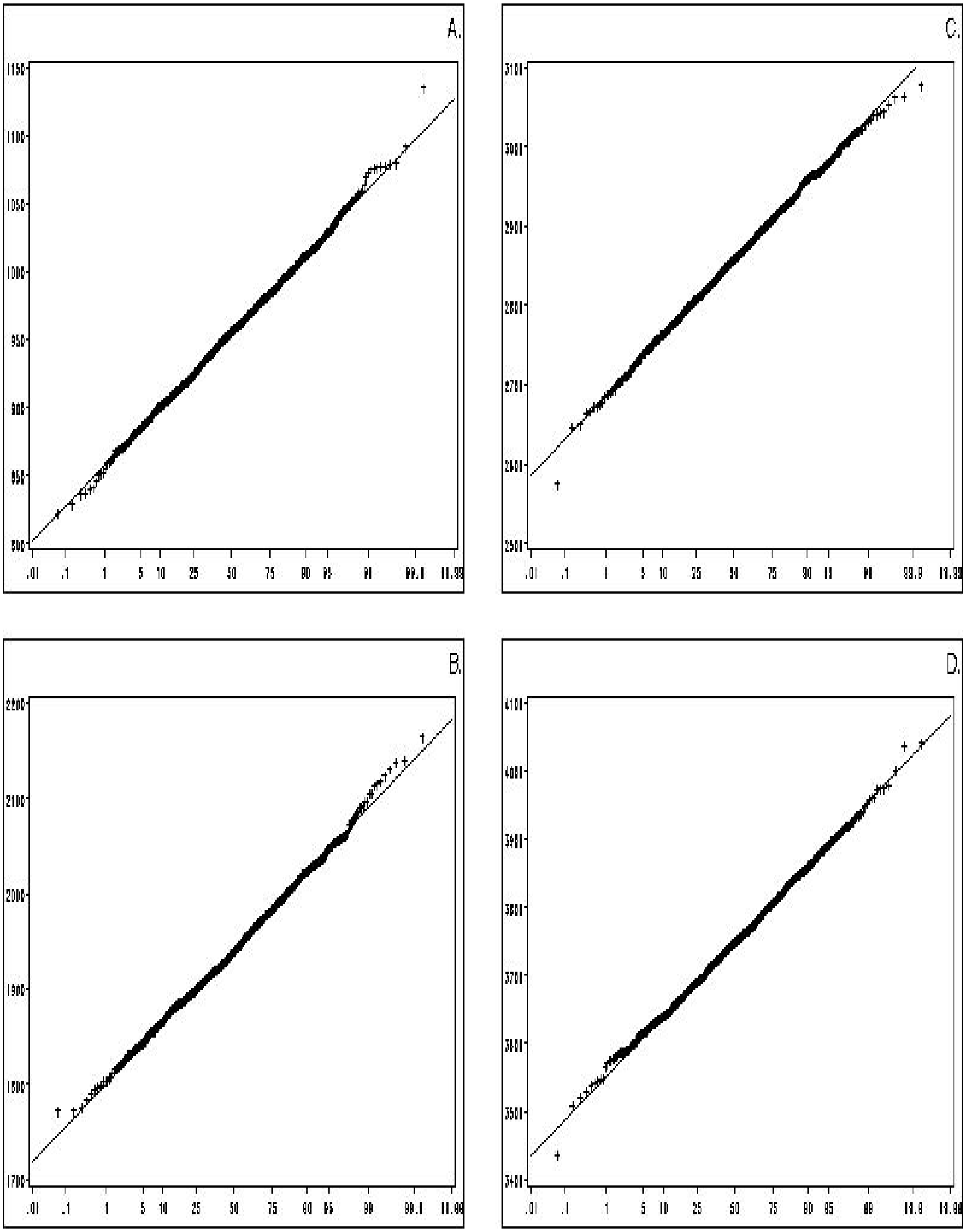

Figure 8: Probability plot for bandwidth=1.0 missing 20%. Horizontal axis represent the theoretical chi-square distribution quantile. Vertical axis represent the observed values. A) 50 subjects B) 100 subjects C) 150 subjects D) 200 subjects

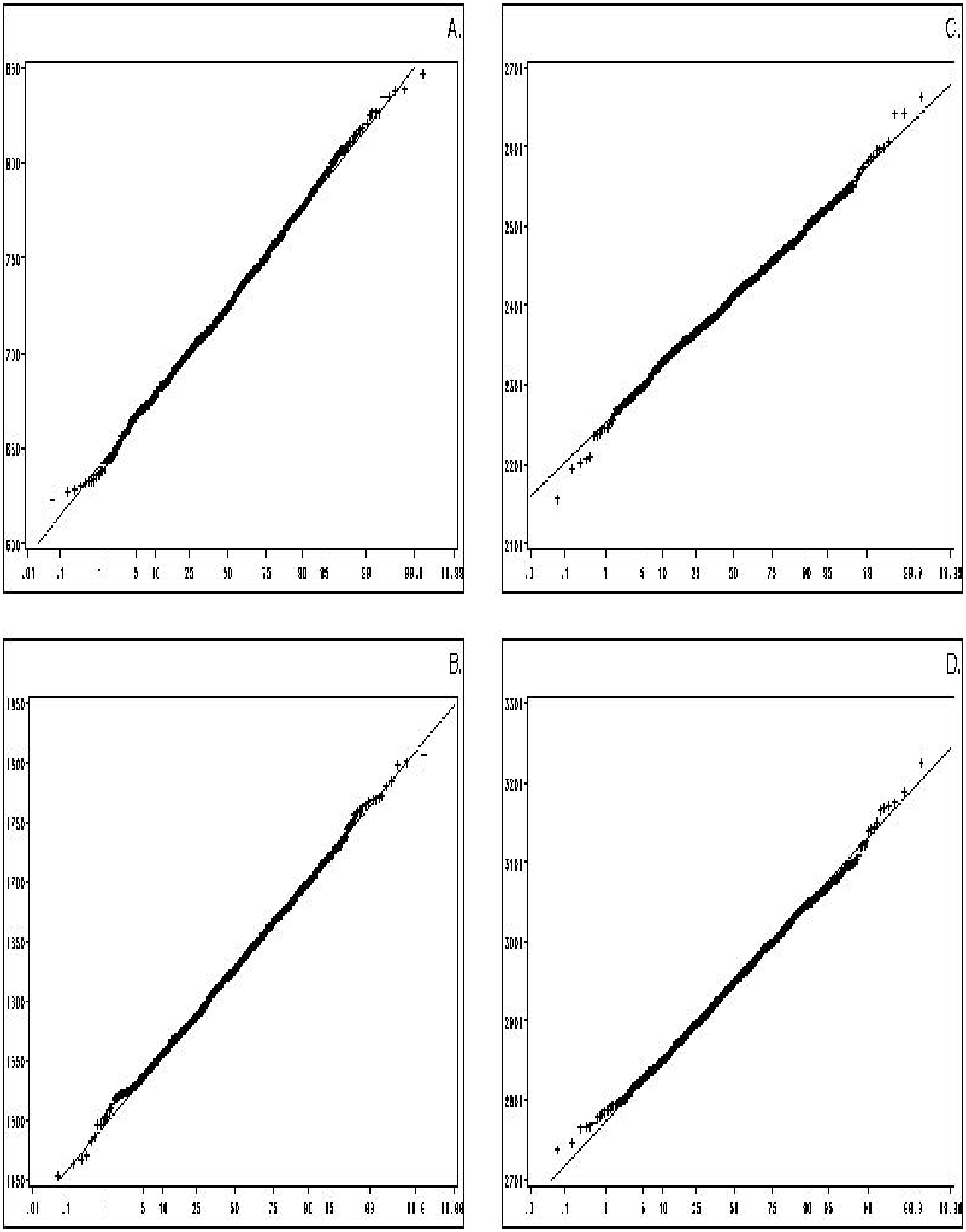

Figure 9: Probability plot for bandwidth=1.0 missing 40%. Horizontal axis represent the theoretical chi-square distribution quantile. Vertical axis represent the observed values. A) 50 subjects B) 100 subjects C) 150 subjects D) 200 subjects

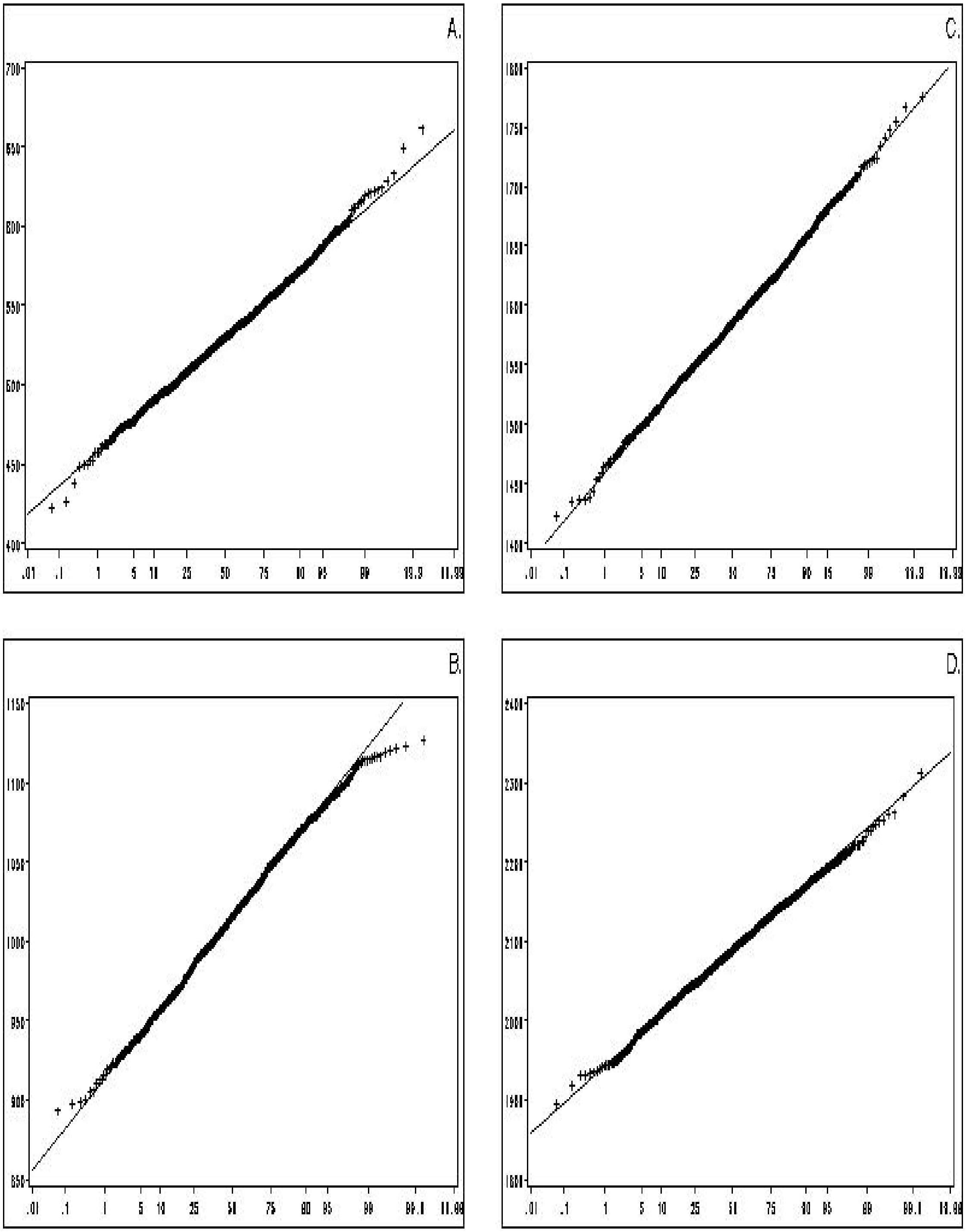

Figure 10: Probability plot for bandwidth=1.0 missing 60%. Horizontal axis represent the theoretical chi-square distribution quantile. Vertical axis represent the observed values. A) 50 subjects B) 100 subjects C) 150 subjects D) 200 subjects

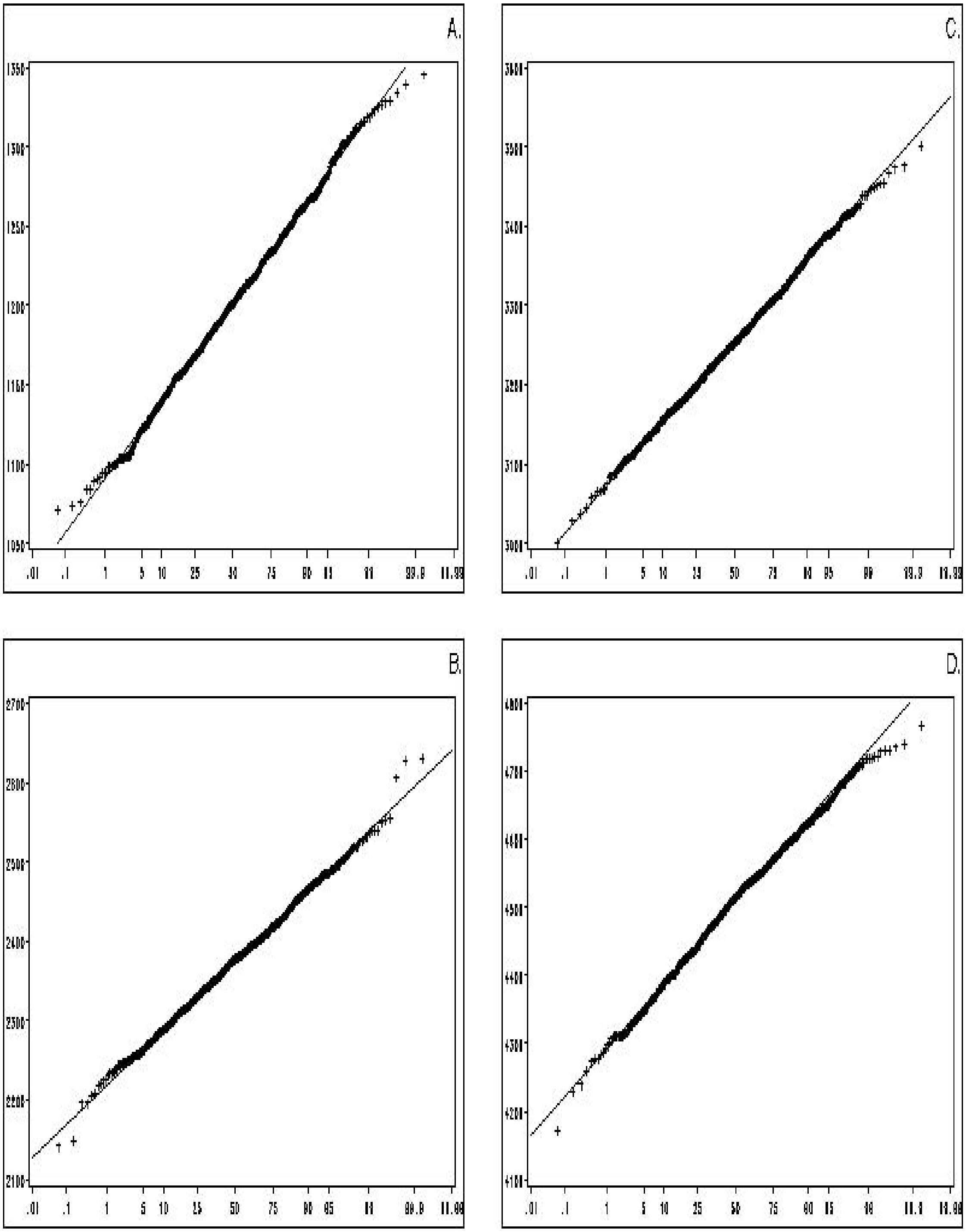

Figure 11: Probability plot for bandwidth=1.5 missing 0%. Horizontal axis represent the theoretical chi-square distribution quantile. Vertical axis represent the observed values. A) 50 subjects B) 100 subjects C) 150 subjects D) 200 subjects

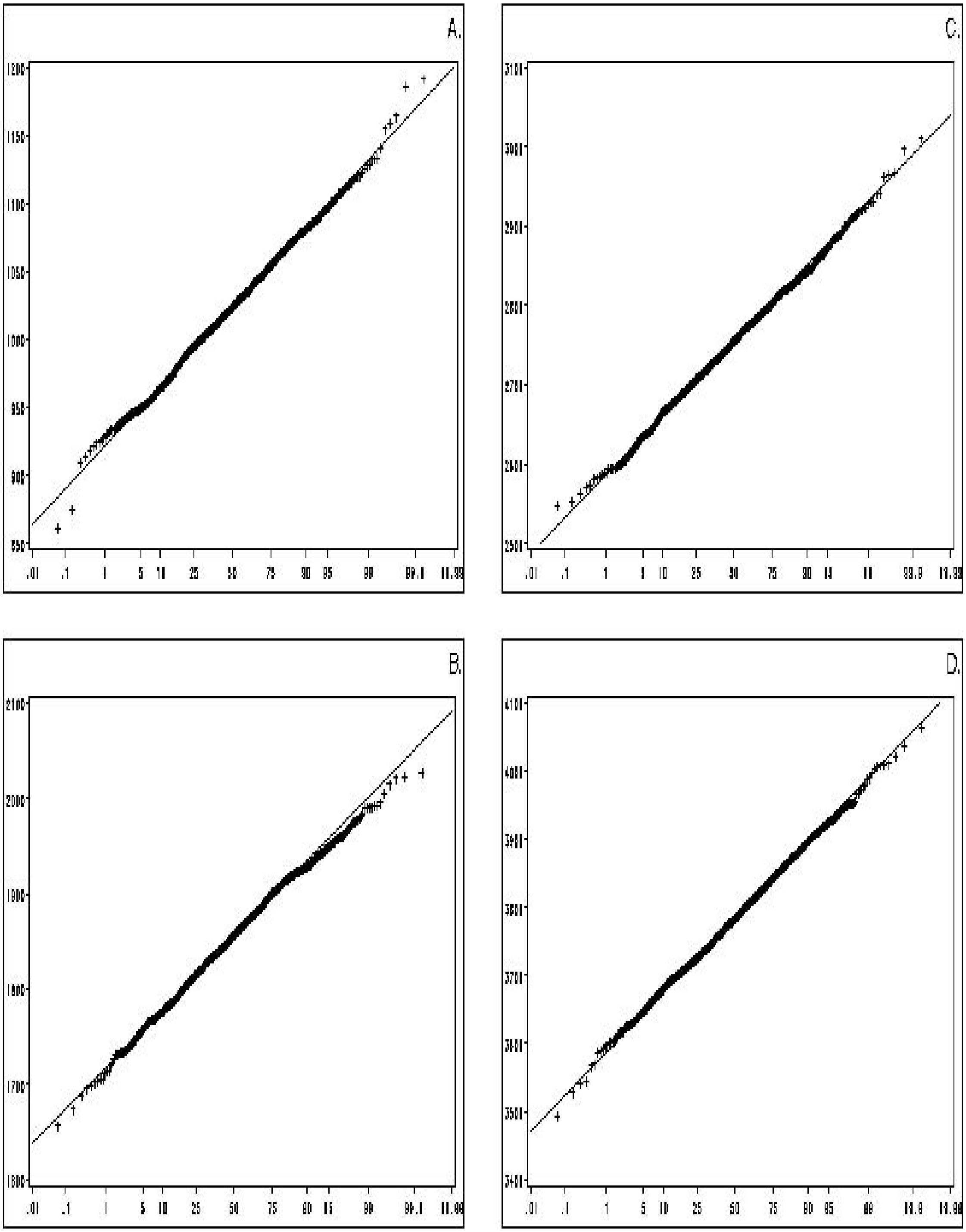

Figure 12: Probability plot for bandwidth=1.5 missing 20%. Horizontal axis represent the theoretical chi-square distribution quantile. Vertical axis represent the observed values. A) 50 subjects B) 100 subjects C) 150 subjects D) 200 subjects

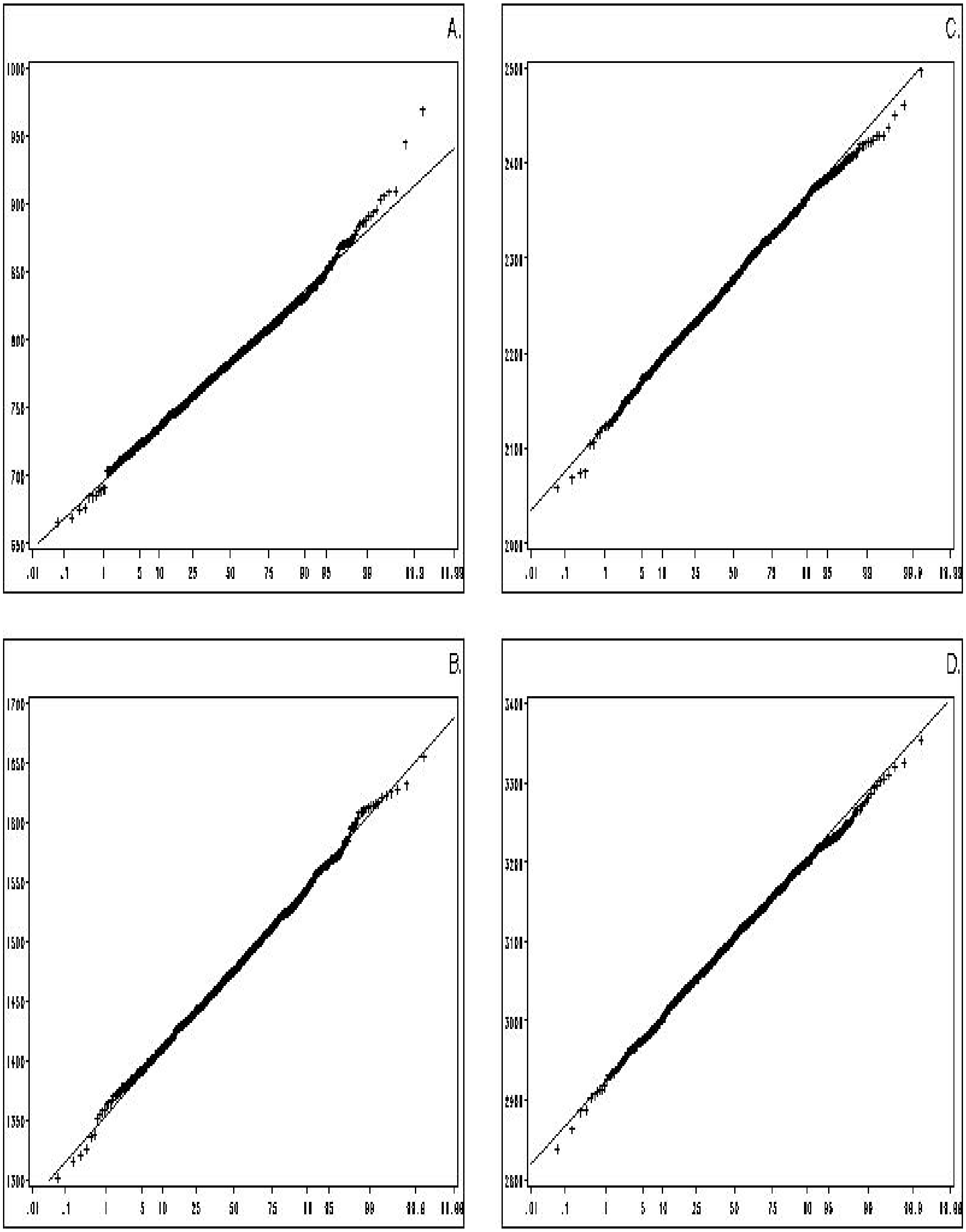

Figure 13: Probability plot for bandwidth=1.5 missing 40%. Horizontal axis represent the theoretical chi-square distribution quantile. Vertical axis represent the observed values. A) 50 subjects B) 100 subjects C) 150 subjects D) 200 subjects

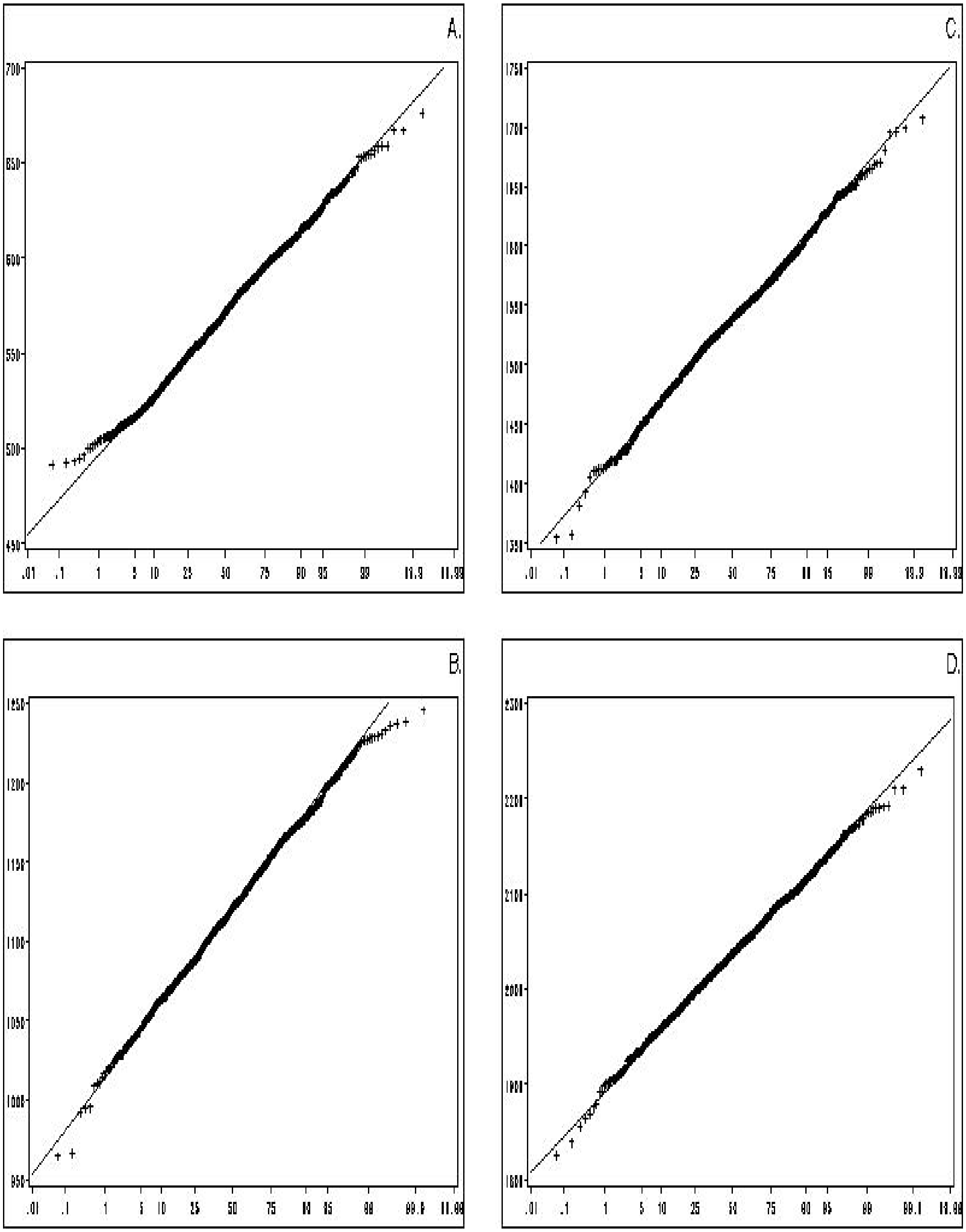

Figure 14: Probability plot for bandwidth=1.5 missing 60%. Horizontal axis represent the theoretical chi-square distribution quantile. Vertical axis represent the observed values. A) 50 subjects B) 100 subjects C) 150 subjects D) 200 subjects

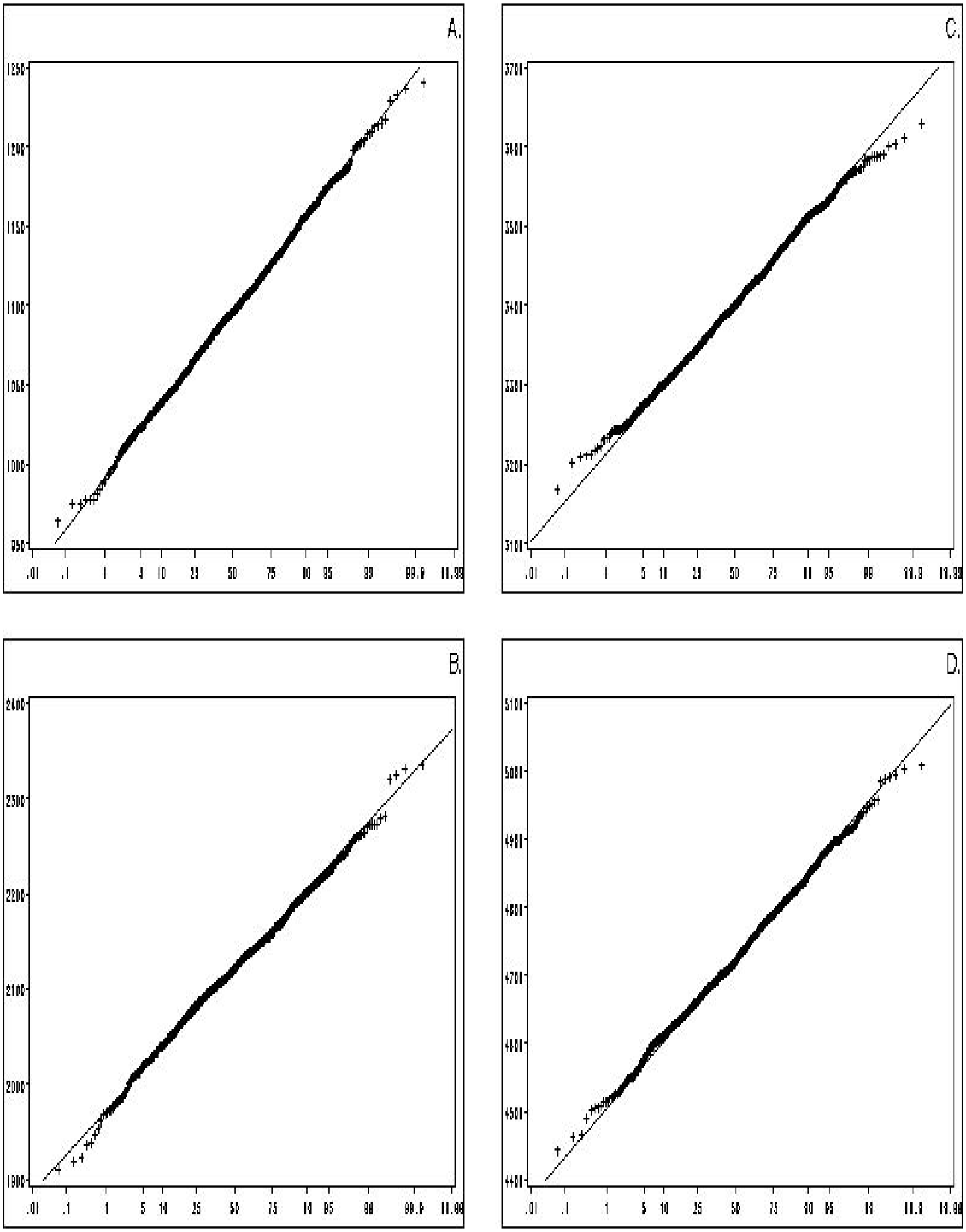

Figure 15: Probability plot for bandwidth=2.0 missing 0%. Horizontal axis represent the theoretical chi-square distribution quantile. Vertical axis represent the observed values. A) 50 subjects B) 100 subjects C) 150 subjects D) 200 subjects

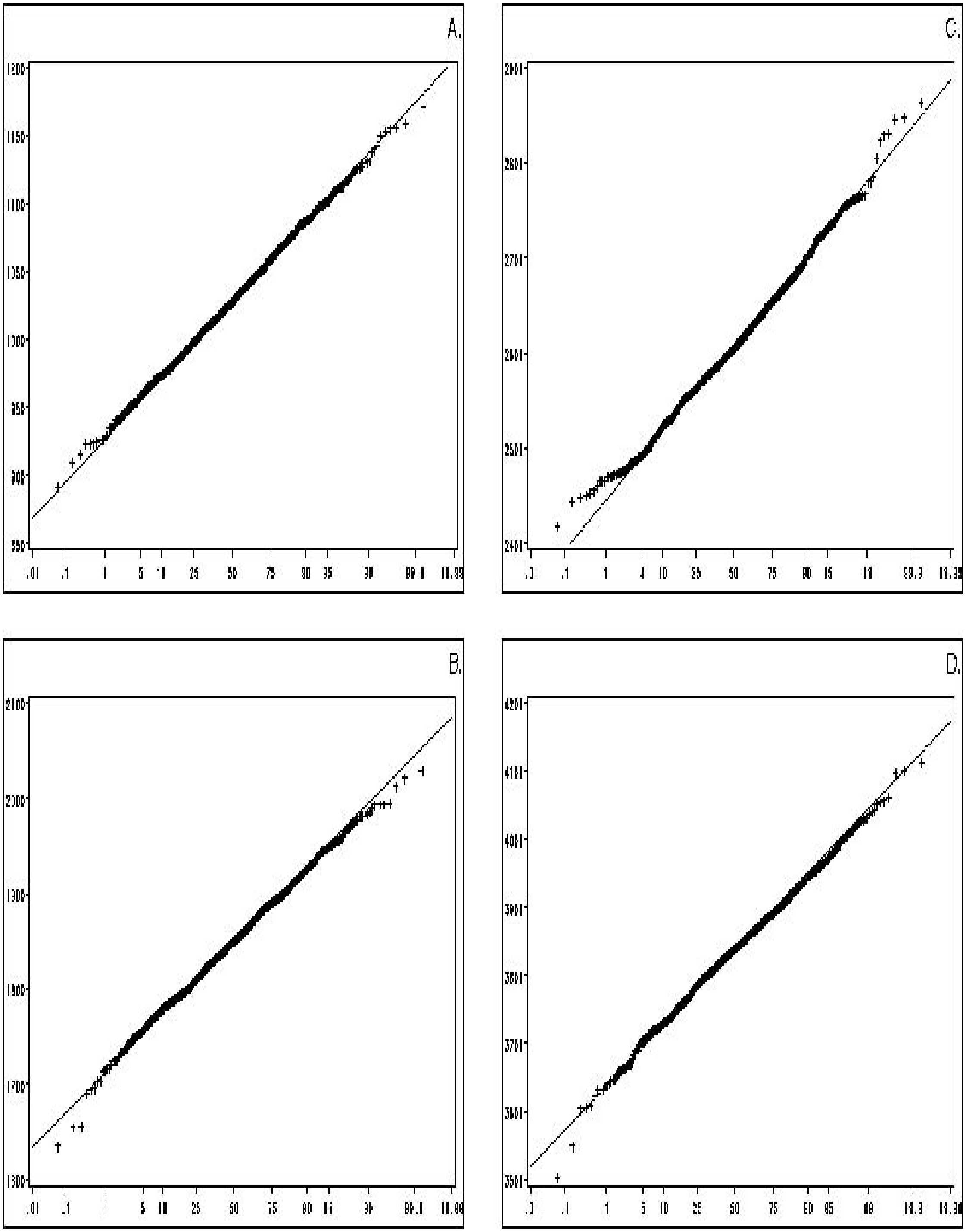

Figure 16: Probability plot for bandwidth=2.0 missing 20%. Horizontal axis represent the theoretical chi-square distribution quantile. Vertical axis represent the observed values. A) 50 subjects B) 100 subjects C) 150 subjects D) 200 subjects

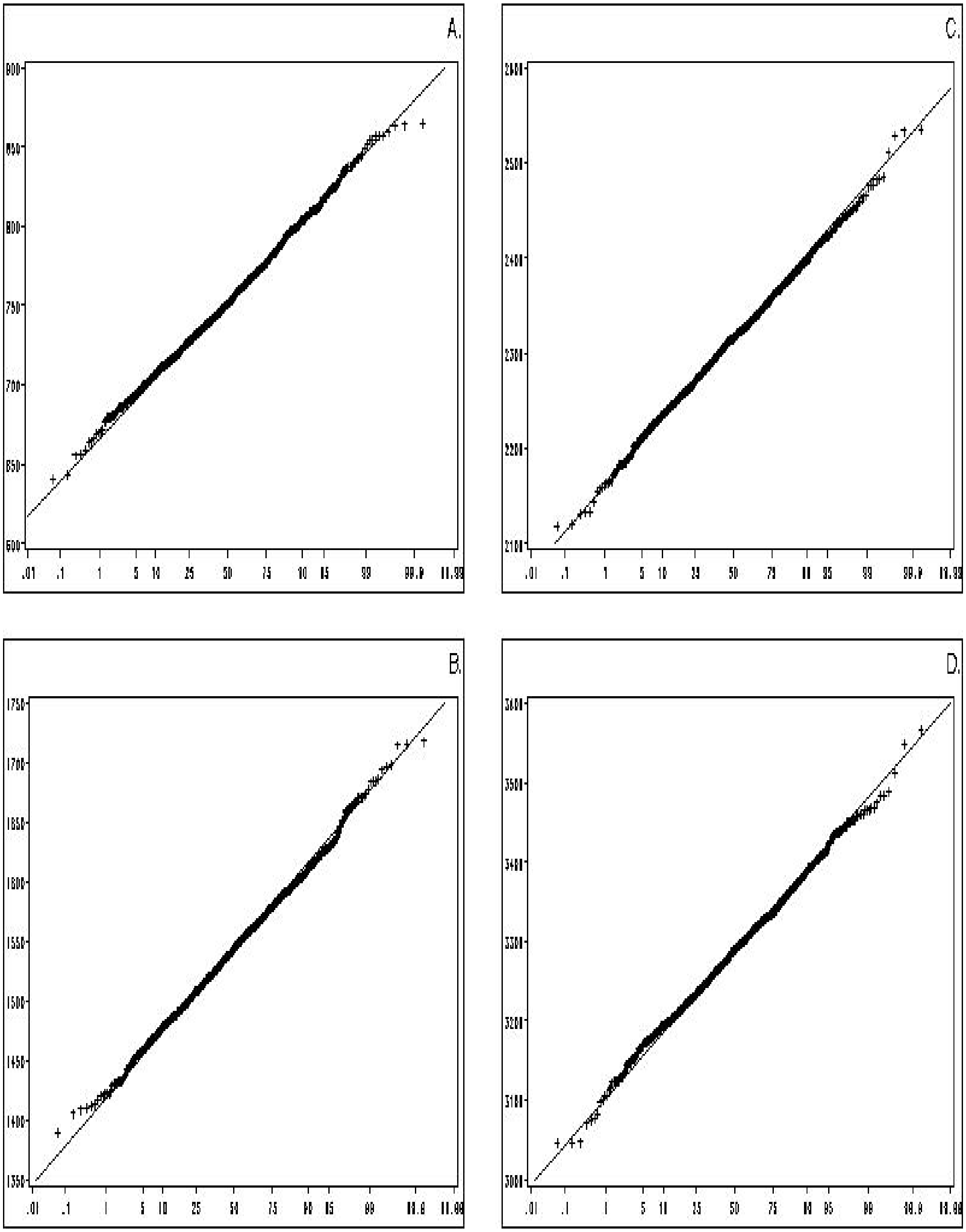

Figure 17: Probability plot for bandwidth=2.0 missing 40%. Horizontal axis represent the theoretical chi-square distribution quantile. Vertical axis represent the observed values. A) 50 subjects B) 100 subjects C) 150 subjects D) 200 subjects

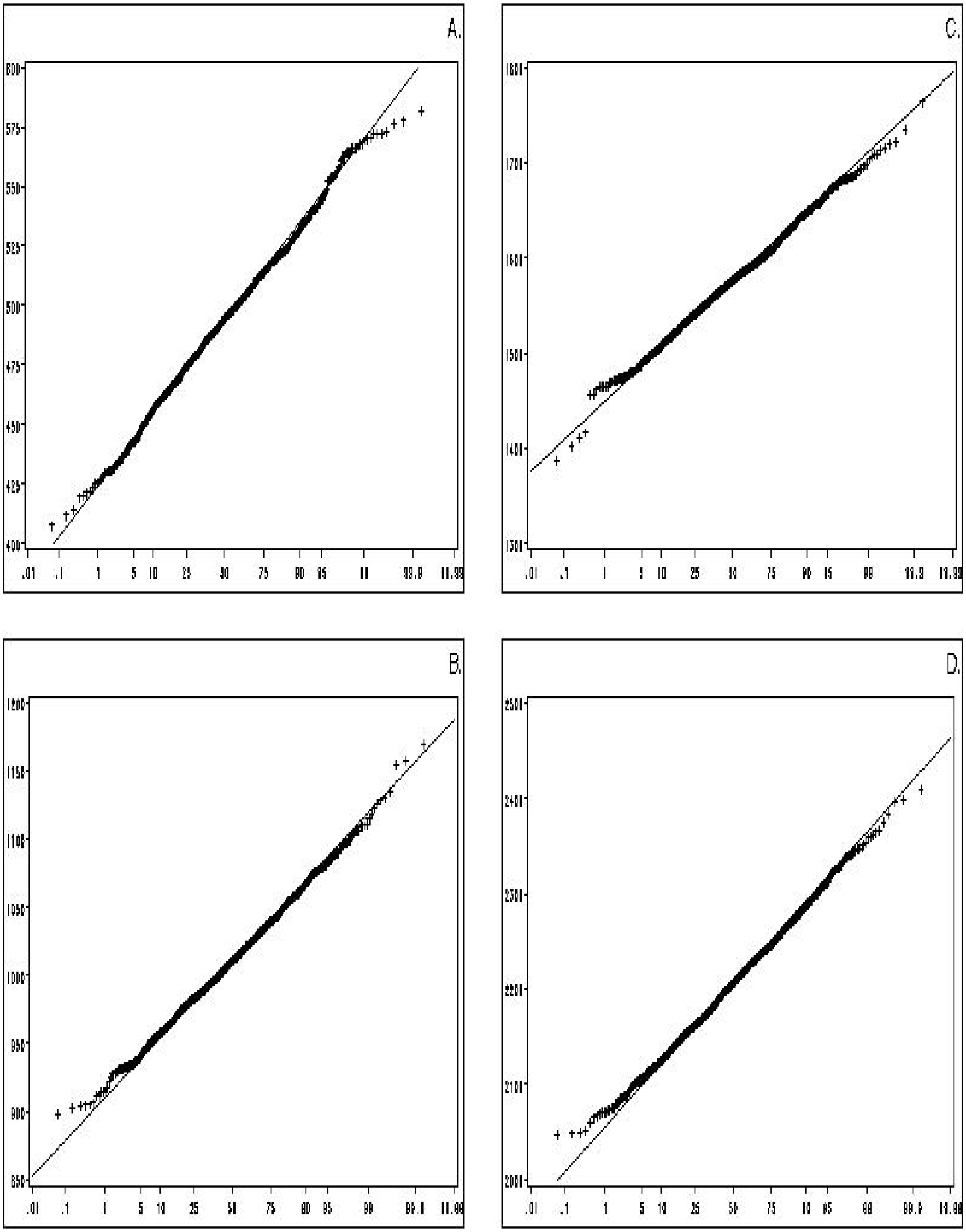

Figure 18: Probability plot for bandwidth=2.0 missing 60%. Horizontal axis represent the theoretical chi-square distribution quantile. Vertical axis represent the observed values. A) 50 subjects B) 100 subjects C) 150 subjects D) 200 subjects

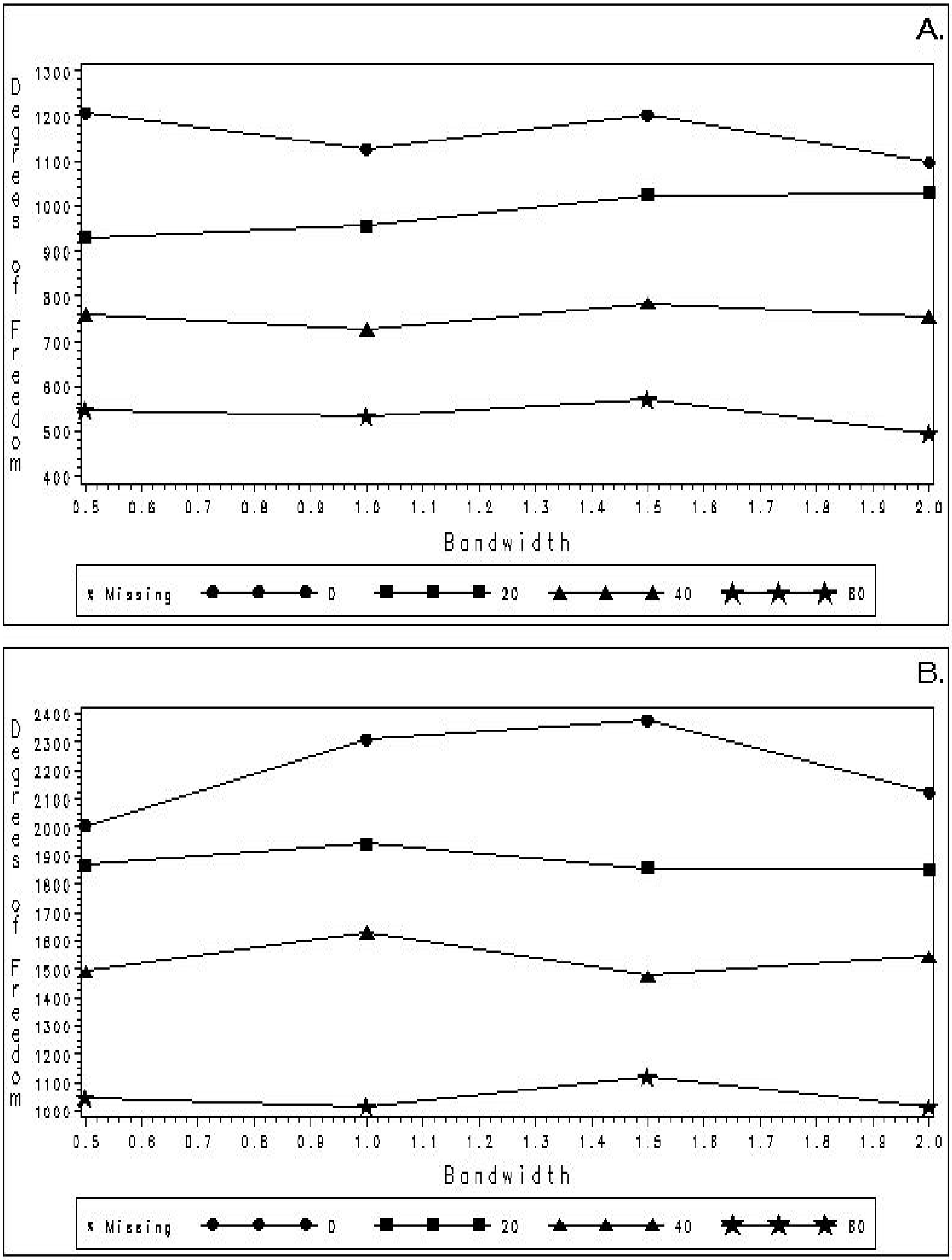

Figure 19: Plot of Degrees of Freedom against bandwidth. A) 50 subjects B) 100 subjects

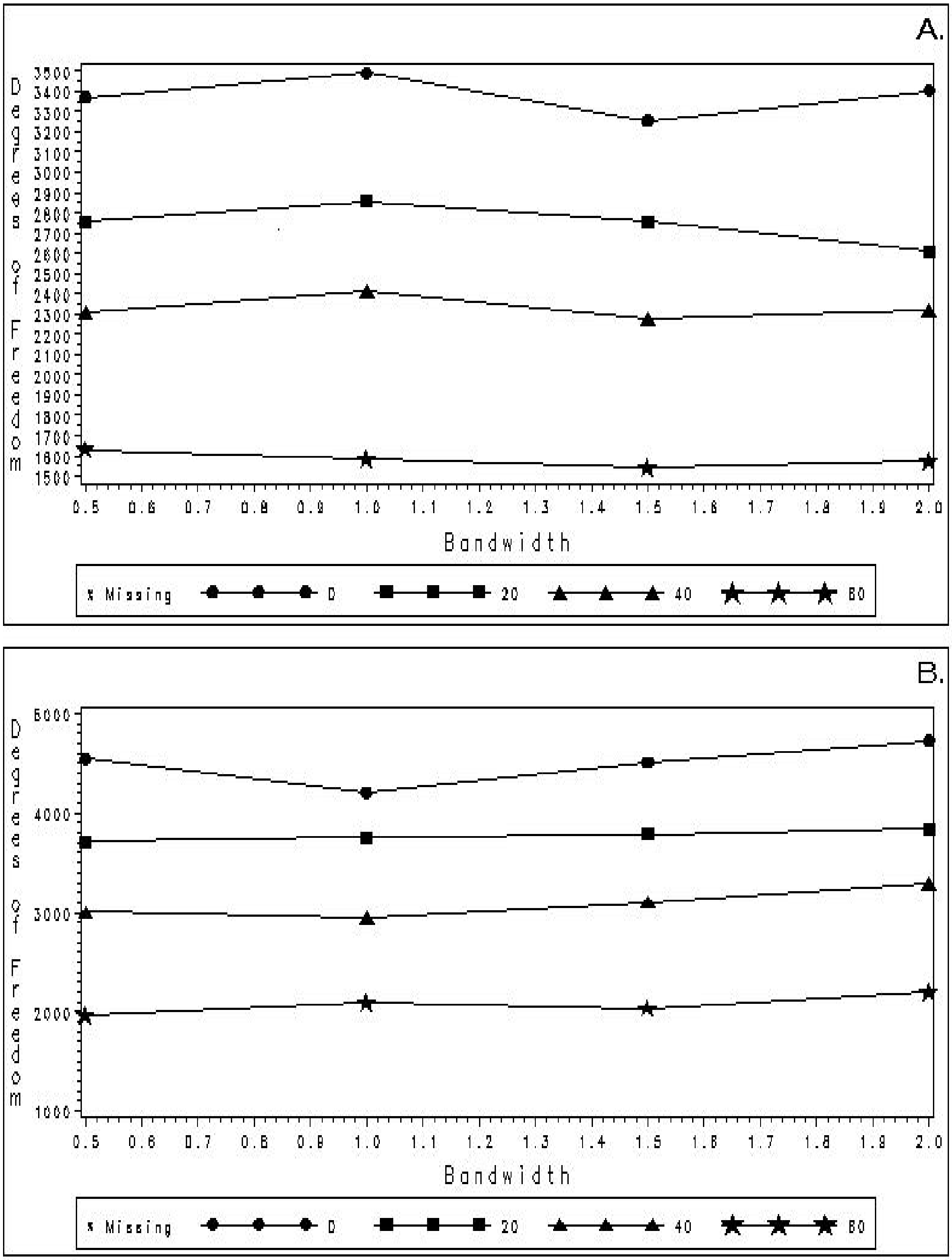

Figure 20: Plot of Degrees of Freedom against bandwidth. A) 150 subjects B) 200 subjects

7 APPENDIX

7.1 FORTRAN CODE FOR MONTE CARLO SIMULATION

```
Program Simulation Use IMSL

Implicit None Integer :: o !Counter and parameter definitions
Integer::i,j,k,n,mi,status,status3,piv,ierror,mii Integer,
Parameter ::covariates=4              !Number of covariate
functions in the model Integer, Parameter :: measurements=30
!Number of time points where measurements are taken Integer,
Parameter :: subjects=100              !Number of subjects in the
study Integer, Parameter :: LDx=4 Integer, Parameter::runtime=1000
Double Precision :: h=1.0,cn=0.0 !Bandwith

Double Precision::BETA0,BETA1,BETA2, BETA3 !Coefficient functions
Double Precision :: gi=0

Double Precision,Dimension(runtime)::var=0 !Variables used to
generate errors

Double Precision, Allocatable, Dimension(:) ::errors !i.i.d.
standard normal errors

Double Precision, Allocatable, Dimension(:):: err !Errors for
chosen structure Double Precision :: variance=4 !Variance of error
process Double Precision, Allocatable, Dimension(:,:) :: covblock
!Covariance of error structure Double Precision, Allocatable,
Dimension(:,:) :: p,pblock !Factor of covariance

!Variables used to generate covariates Double Precision,
Dimension(covariates,subjects) :: x=1 !Matrix with columns
corresponding to

!covariate values for each subject Integer value(subjects)
!Holders for randomly generated covariates

Double Precision value2(subjects)

Double Precision, Dimension(measurements,covariates,subjects) ::
design=0

    !Variables used to generate measurement times
Double Precision, Dimension(measurements+1,subjects) :: t=0 !Cols
are measurment times per subject

Double Precision misses(measurements) !Missing data indicators
Integer, Dimension(subjects) :: m=measurements !Number of
observations per subject

!Simulated observations

Double Precision,Dimension(measurements,subjects) :: y=0   !Cols
are observations on a subject.
```

```
Double Precision,Dimension(measurements,subjects) :: Eij=0

Double Precision :: Errorsq=0

Double Precision,Dimension(measurements,subjects) :: ye=0  ! y
expected Double Precision, Dimension(90) :: g=0 !Grid matrix
Double Precision, Dimension(measurements,measurements) :: Kernel=0
!Kernel matrix Double Precision :: Pi=3.14159

Double Precision, Dimension(measurements,covariates) :: Xi=0  !X
Matrix for ith subject

Double Precision, Dimension(measurements) :: Yi !Y Vector for ith
subject Double Precision, Dimension(covariates,measurements) ::
Xtk   ! XTranspose by Kernel

Double Precision,Dimension(covariates,covariates) :: Xkx     !
XTranspose by Kernel by Xi

Double Precision,Dimension(covariates,covariates) :: Xkxtotal
Double Precision, Dimension(covariates,covariates) :: Xinv

Double Precision, Dimension(covariates) :: Xky ! Xtranspose by
Kernel by Yi Double Precision, Dimension(covariates) :: Xkytotal
Double Precision, Dimension(covariates,measurements,subjects) ::
Bt=0 ! B matrix for t
```

```
Double Precision, Dimension(covariates) :: BB

!Other variables Integer :: total !Total number of observations
Integer :: position

!Time point generation Do i=1, subjects
    t(1,i)=DRNUNF()          !Generate initial time points!
    Do j=1, measurements
    t(j,i)=t(1,i)+(j-1)  !Generate remaining "scheduled" time points
    End Do
End Do

Do i=1, subjects
    m(i)=30
    Call DRNUN(measurements, misses) !Generate random "missing
    indicators"
    Do j=measurements, 2, -1
    If (misses(j) .lt. 0.4) Then
    m(i)=m(i)-1                  !Update number of observations
    Do k=j, measurements
    t(k,i)=t(k+1,i)              !Remove missing observations
    End Do
    End If
    End Do
End Do !End of time point generation

!Compute total number of observations
```

```
total=0 Do i=1, subjects
    total=total+m(i)
End Do

Do i=1,subjects

Open(unit=30,file=’variance.dat’,status=’replace’,iostat=ierror)
write(30,*)  t(:,i) write(30,*) ’ ’ End do

Do o=1,runtime gi=0 var=0 x=1 design=0 y=0 Eij=0 Errorsq=0 ye=0
g=0 Kernel=0 Xi=0 Bt=0

!!!!!Begin generation of data!!!!!!!!!!!!!

Call RNSET (0) !Sets seed to system clock Do k=2, covariates-1
    Call RNBIN(subjects, 1, .5, value)
    !Generate random binary covariates
    x(k,1:subjects)=value
End Do Call DRNNOA(subjects, value2)
    x(covariates,1:subjects)=.5*value2 !Generate normal covariates
!Call DWRRRN(’ ’, covariates, subjects, x, covariates, 0)

!Build errors Allocate (errors(total), STAT=status)

!Allocate space for i.i.d. errors
```

```
Call DRNNOA(total, errors)    !Generate i.i.d std. normal errors

Allocate (p(total,total), STAT=status3)  !Allocate space for
factored covariance str.

position=0 Do i=1, subjects
    Allocate (covblock(m(i),m(i)),pblock(m(i),m(i)))
    covblock=0.0
    pblock=0.0
    Do j=1, m(i)
        Do k=1, m(i)
        covblock(j,k)=variance*DEXP(-1*DABS(t(j,i)-t(k,i)))
        End Do
    End Do
    Call DCHFAC (m(i), covblock, m(i), 100*DMACH(4), piv,
     pblock, m(i))
    p(position+1:position+m(i),position+1:position+m(i))=
    pblock(1:m(i),1:m(i))
    position=position+m(i)
    Deallocate (covblock,pblock)
End Do

Allocate (err(total))      !Allocate space for actual errors

    !Compute errors from factorization and i.i.d errors
Call DMURRV (total, total, p, total, total, errors, 2, total, err)
    !End of error build
    !Build simulated observations
position=0 Do i=1, subjects
    Do j=1, m(i)
       y(j,i)=BETA0(t(j,i))+BETA1(t(j,i))*x(2,i)+BETA2(t(j,i))*x(3,i)
        +BETA3(t(j,i))*x(4,i)+err(position+j)
    End Do
    position=position+m(i)
End Do

!!!!!End of data generation!!!!!!! Write (*,*) o,’done’

!Build design matricies for each subject Do i=1, subjects
    Do j=1, m(i)
        design(j,:,i)=x(:,i)
    End Do
End Do

!Call DWRRRN (’y’, measurements, subjects, y, measurements, 0)

Do j=1,subjects
   mii=m(j)
   Do i= 1,mii
   gi=t(i,j)
   call A(design,subjects,measurements,covariates,Xkxtotal,
     gi,Pi,h,Kernel,m,t)
   call B(design,y,subjects,measurements,covariates,Xkytotal,
     gi,Pi,h,Kernel,m,t)
   CALL DLINRG (covariates, Xkxtotal, covariates, Xinv, covariates)
   CALL DMRRRR (covariates, covariates, Xinv, covariates, covariates,
     1, Xkytotal, covariates, covariates, 1, BB, covariates)
     Do n=1,covariates ! Calculation for Bt by Xkx(inverse) by Xky
         Bt(n,i,j)=BB(n)
      End do
   End do
End do

Errorsq=0

Do i=1,subjects
    Do j=1,m(i)
    ye(j,i)=Bt(1,j,i)+Bt(2,j,i)*x(2,i)+Bt(3,j,i)*x(3,i)+Bt(4,j,i)
    *x(4,i)
    Eij(j,i)=(y(j,i)-ye(j,i))**2
    Errorsq=Errorsq+Eij(j,i)
    end do
End do

var(o)=Errorsq/(total-subjects)

Open (unit=30,file=’variance.dat’,status=’replace’,iostat=ierror)
write(30,*)  var(o) Errorsq=0 Deallocate (errors, STAT=status)
Deallocate (p, STAT=status3) Deallocate (err)
!Deallocate(covblock,pblock)
```

```
End do End Program




!!!!!!!!!!!!!!!!!!!Subroutines and functions!!!!!!!!!!!!!!!!!!!!

!Coefficient functions Double Precision Function BETA0(s)
    Implicit None
    Double Precision, Intent(IN) :: s
    Double Precision, Parameter :: pi=3.14159265359
    BETA0 = 15 + 20*DSIN(s*pi/60)
End Function

Double Precision Function BETA1(s)
    Implicit None
    Double Precision, Intent(IN) :: s
    BETA1 = 4 - ((s - 20)/10)**2
End Function

Double Precision Function BETA2(s)
    Implicit None
    Double Precision, Intent(IN) :: s
    Double Precision, Parameter :: pi=3.14159265359
    BETA2 = 2 - 3*DCOS((s-25)*pi/15)
End Function
```

```
Double Precision Function BETA3(s)
    Implicit None
    Double Precision, Intent(IN) :: s
    BETA3 = -5 + (30 - s)**3/5000
End Function !End of coefficient functions



!Subroutine for creat Kernel matrix



subroutine Kernelmatrix(Kernel,gi,Pi,t,h,j,
mi,measurements,subjects) integer :: measurements,subjects

Double Precision, Dimension(measurements,measurements) :: Kernel
Double Precision, Dimension(measurements+1,subjects) :: t

Double Precision :: gi,Pi,h integer :: j,k,mi

Do k=1,mi
    Kernel(k,k)=(1.0/(h*(sqrt(2.0*Pi))))*(exp(-((gi-t(k,j))**2)/
    (2.0*(h**2))))       !Value Kernel matrix For each subject
end do return End subroutine

!Subsoutine for Xi transpose by K by Xi

subroutine A(design,subjects,measurements,
covariates,Xkxtotal,gi,Pi,h,Kernel,m,t)
```

```
Integer :: j,k,subjects,measurements,covariates,mi

Double Precision, Dimension(measurements,covariates,subjects) ::
design Double Precision, Dimension(measurements,measurements) ::
Kernel Double Precision, Dimension(measurements+1,subjects) :: t
Integer, Dimension(subjects) :: m

Double Precision, Dimension(measurements,covariates) :: Xi

Double Precision, Dimension(covariates,measurements) :: Xtk

Double Precision, Dimension(covariates,covariates) :: Xkx

Double Precision, Dimension(covariates,covariates) :: Xkxtotal
Double Precision :: gi,Pi,h Xkxtotal=0

Do j=1,subjects
   mi=m(j)
   call Kernelmatrix(Kernel,gi,Pi,t,h,j,mi,measurements,subjects)
   Xi(:,:)=design(:,:,j)
   CALL DMXTYF (measurements, covariates, Xi, measurements,
   measurements, measurements, Kernel, measurements, covariates,
   measurements, Xtk, covariates)
   ! Multiply Xi transpose by Kernel matrix
   CALL DMRRRR (covariates, measurements, Xtk, covariates,
   measurements, covariates,
   Xi, measurements, covariates,covariates, Xkx, covariates)
   ! Multiply Xi by kernel by Xi
   Do k=1, covariates
       Xkxtotal(k,:)=Xkxtotal(k,:)+Xkx(k,:) !Sum of Xkx for all subject
   End do
End do return end subroutine

!Subsoutine for Xi transpose by K by Yi

subroutine (design,y,subjects,measurements,
covariates,Xkytotal,gi,Pi,h,Kernel,m,t)

Integer :: j,k,subjects,measurements,covariates,mi

Double Precision, Dimension(measurements,covariates,subjects) ::
design

Double Precision, Dimension(measurements,subjects) :: y

Double Precision, Dimension(measurements,measurements) :: Kernel

Double Precision, Dimension(measurements+1,subjects) :: t

Integer,Dimension(subjects) :: m

Double Precision, Dimension(measurements,covariates) :: Xi
Double Precision,Dimension(measurements) :: Yi

Double Precision,Dimension(covariates,measurements) :: Xtk

Double Precision, Dimension(covariates) :: Xky

Double Precision,Dimension(covariates) :: Xkytotal

Double Precision :: gi,Pi,h Xkytotal=0

Do j=1,subjects
   mi=m(j)
   call Kernelmatrix(Kernel,gi,Pi,t,h,j,mi,measurements,subjects)
   Xi(:,:)=design(:,:,j)
   Yi(:)=y(:,j)
   CALL DMXTYF (measurements, covariates, Xi, measurements,
   measurements,measurements, Kernel, measurements, covariates,
   measurements, Xtk, covariates)
   ! Multiply Xi transpose by Kernel matrix
   CALL DMURRV (covariates, measurements, Xtk, covariates,
   measurements, Yi, 1,covariates, Xky) !Multiply Xi by kernel by Yi
   Do k=1, covariates
      Xkytotal(k)=Xkytotal(k)+Xky(k)    ! Sum of Xky for all subjects
   End do
End do return end subroutine
```

7.2 SAS CODE

```
 libname dat ’C:\simulation data library’;
 title ’H15\_n150\_m60’;

proc univariate data=dat.H15\_n150\_m60;
    var variance;
    output out=result var=estvar ;
run;

data new;
    set result;
    k=32/estvar;
    alpha=k/2;
    dummy=1;
run;

data new2;
    set dat.H15\_n150\_m60;
    dummy=1;
run;

data new3 (keep=chi);
    merge new new2;
    by dummy;
    chi=k*variance/4;
run;
```

```
proc capability;
    probplot chi/gamma(alpha=est sigma=2 theta=0);
run;

proc univariate data=new3;
    var chi;
    output out=result2 std=eststd ;
run;

data new4;
   set result2;
   E\_H15\_n150\_m60=(1.96*eststd)/sqrt(1000);
run;

proc print data=new4;
    var E\_H15\_n150\_m60;
    var eststd;
run;
```