

**INSTITUTO TECNOLÓGICO Y DE ESTUDIOS
SUPERIORES DE MONTERREY**

CAMPUS MONTERREY

**PROGRAMA DE GRADUADOS DE LA DIVISION DE
ELECTRONICA, COMPUTACION, INFORMACION
Y COMUNICACIONES**



**INTERVALOS DE CONFIANZA PARA LA
DISPONIBILIDAD DE UN SISTEMA MEDIANTE
BOOTSTRAP**

TESIS

**PRESENTADA COMO REQUISITO PARCIAL
PARA OBTENER EL GRADO ACADÉMICO DE:
MAESTRO EN CIENCIAS
ESPECIALIDAD EN ESTADÍSTICA APLICADA**

TANIA GUADALUPE RAMIREZ GUERRA

JUNIO DE 2002

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE
MONTERREY
Campus Monterrey

DIVISION DE ELECTRÓNICA, COMPUTACIÓN, INFORMACIÓN Y
COMUNICACIONES
PROGRAMA DE GRADUADOS



INTERVALOS DE CONFIANZA PARA LA DISPONIBILIDAD
DE UN SISTEMA MEDIANTE BOOTSTRAP

TESIS

PRESENTADA COMO REQUISITO PARCIAL PARA
OBTENER EL GRADO ACADÉMICO DE:

MAESTRO EN CIENCIAS
ESPECIALIDAD EN ESTADÍSTICA APLICADA

TANIA GUADALUPE RAMIREZ GUERRA

JUNIO DE 2002

Dedicatoria

A **Dios** por darme la oportunidad de vivir y lograr una meta más.

A mi papá **Luciano** (Q. E. P. D.) por el amor y todas las enseñanzas que me ha dejado.

A mi mamá **Avelly** por su amor incondicional, comprensión, ejemplo, apoyo y enseñanza. Gracias ma.

A mi hermano **Raúl Asbury** por el amor que nos une, gracias por el apoyo y cariño que me ha brindado.

Y muy especialmente a Mi Hija **Lucianny Guadalupe** por ser la luz y la estrella que ilumina mi vida.

Agradecimientos

Especialmente a mi asesor el **Dr. José Guadalupe Ríos Alejandro** por su enseñanza, tiempo y sabios consejos.

A mis sinodales, **Dr. Christian Geraud Garrigoux Michel** y **Dra. Rebeca Romero Alvarez** por su tiempo y consejos.

A **Guillermo Arturo Álvarez Pérez** por sus comentarios muy atinados y su apoyo.

A la **Ing. Sandra P. González Lozano** por su apoyo en todo momento y por creer en mí.

De igual forma, a todos y cada uno de las personas que han contribuido a la elaboración de este trabajo de tesis.

A todos ellos, Muchas Gracias

Indice

Resumen

Capítulo 1	Introducción.....	1
	1.1 Objetivo	5
	1.2 Metodología.....	6
	1.3 Estructura de la tesis.....	7
Capítulo 2	Antecedentes.....	8
	2.1 Disponibilidad.....	8
	2.2 Intervalos de confianza en el caso Exponencial – Exponencial.....	12
	2.2. 1 Ejemplo del Intervalo de confianza para el caso Exponencial – Exponencial.....	14
	2.3 Método Wang- Kececioglu (WK).....	16
	2.3.1 Ejemplo del método Wang- Kececioglu.....	19
Capítulo 3	Método Bootstrap.....	22
	3.1 Método Bootstrap.....	22
	3.2 Procedimiento del Método Bootstrap.....	23
	3.3 Empleo del método Bootstrap estándar en la	

	estimación de Intervalos de confianza para la disponibilidad.....	26
	3.4 Representación de la simulación Bootstrap.....	28
Capítulo 4	Estudio de Desempeño.....	29
	4.1 Descripción de la simulación.....	29
	4.2 Funciones de probabilidad de variables aleatorias continuas.....	33
	4.3 Desplazamiento y escalamiento de las muestras aleatorias.....	41
	4.4 Equivalencia entre simulaciones.....	43
	4.5 Validación del código de simulación: Ejemplo de la distribución Exponencial.....	45
	4.6 Prueba para medir el porcentaje de cobertura.....	47
Capítulo 5	Conclusiones.....	49
	5.1 Trabajo a futuro.....	52
Anexos		
	Anexo A: Validación del código de simulación: Ejemplo de la distribución Exponencial.....	53
	Anexo B: Tablas de la simulación.....	57

Anexo C: Tablas de igualdad de proporciones.....	64
Anexo D: Código del programa.....	73

Referencias Bibliográficas.....	88
--	-----------

Vita.....	90
------------------	-----------

Lista de figuras

3.1 Proceso para estimar el intervalo de confianza del estadístico $\hat{\theta}$, mediante Bootstrap.

4.1 Distribuciones utilizadas para los tiempos entre fallas.

4.2 Distribuciones utilizadas para los tiempos de reparación.

4.3 Parámetros y distribuciones empleadas.

4.4 Distribución normal estándar o Normal($\mu=0, \sigma^2=1$).

4.5 Distribución Exponencial ($\beta=1$).

4.6 Distribución t de student($\nu=4$).

4.7 Distribución Lognormal($\mu=0, \sigma^2=1$).

4.8 Distribución Gamma($\alpha=2, \beta=1$).

4.9 Distribución Weibull($\alpha=2, \beta=1$).

4.10 Número de tablas resultantes de la simulación.

Resumen

La disponibilidad es un concepto muy importante para el usuario de productos y sistemas reparables, esta nos indica el porcentaje de tiempo que se espera que los sistemas funcionen satisfactoriamente [Wang y Kececioglu (2000)]. La disponibilidad más usada es la disponibilidad inherente [Ireson, (1996)]. Existe una forma de acercarse al problema de precisión de la estimación del parámetro de disponibilidad inherente a través de la construcción de intervalos de confianza. Los límites de confianza exactos sobre la disponibilidad inherente están determinados solo cuando los tiempos entre fallas y los tiempos de restauración se distribuyen exponencialmente [Wang y Kececioglu (2000)] y, como se sabe, en procesos reales no siempre estos tiempos se distribuyen exponencialmente. En el año 2000 Wang y Kececioglu proponen un método para la estimación de dichos intervalos de confianza. En este trabajo se estudia el desempeño de la técnica no paramétrica Bootstrap estándar para la estimación de Intervalos de confianza para la disponibilidad inherente y se lleva a cabo una comparación entre las estimaciones obtenidas mediante este método y el método propuesto por Wang y Kececioglu (2000). Mediante una simulación realizada para dicha comparación se observa que el método Bootstrap estándar tiene mejor desempeño que el método propuesto por Wang y Kececioglu.

Capítulo 1

Introducción

En una planta, la confiabilidad de sistemas es muy importante. Relativamente pocos sistemas son diseñados para operar sin un buen mantenimiento. El mantenimiento puede ser preventivo o correctivo. El mantenimiento preventivo es realizado antes de que la falla haya ocurrido y el objetivo es incrementar la confiabilidad durante un largo periodo de tiempo, el mantenimiento correctivo es realizado después de que la falla ha ocurrido, y de esta forma regresar el sistema a un buen funcionamiento tan pronto como sea posible, el objetivo del mantenimiento correctivo es maximizar la disponibilidad [Lewis, (1987)]. El principal criterio para juzgar el mantenimiento preventivo es el aumentar la confiabilidad, pero es necesario tener un criterio diferente para juzgar la eficiencia del mantenimiento correctivo y el criterio más usado es la disponibilidad de sistemas, la cual es definida robustamente, como la probabilidad de que el sistema se encuentre en condiciones de funcionamiento cuando este sea requerido [Lewis, (1987)].

La disponibilidad de un sistema o equipo es la probabilidad de que el sistema o equipo se encuentren funcionando satisfactoriamente en cualquier

punto en el tiempo cuando existen condiciones dadas. La disponibilidad es un término muy usado en la industria. La disponibilidad puede ser declarada en términos aplicables a datos específicos, por ejemplo en el diseño o en el empleo operacional. Y ésta puede ser definida como *la disponibilidad intrínseca o inherente* la cual considera operación y actividad de un sistema bajo un período. *La disponibilidad operacional* toma los mismos términos que la disponibilidad inherente, excepto que el término: “tiempo de reparación” es sustituido por el término: “tiempo total de reparación” y por último *la disponibilidad usada* considera el tiempo que está fuera de funcionamiento el sistema [Ireson, (1966)].

Por lo tanto, la disponibilidad es un concepto muy importante para el usuario de productos o sistemas reparables, ya que nos indica el porcentaje de tiempo que se espera que los productos funcionen satisfactoriamente.

La fórmula para calcular una estimación puntual de la disponibilidad inherente es conocida, ya que involucra medias poblacionales del tiempo entre fallas y el tiempo de reparación. La disponibilidad más usada es la disponibilidad intrínseca o inherente [Ireson, (1966)] y es definida por:

$$(1.1) \quad A = \frac{\mu_{Tf}}{\mu_{Tf} + \mu_{Tr}}$$

donde:

μ_{Tf} = media de los tiempos entre fallas (MTBF)

μ_{Tr} =media de los tiempos en restaurar el sistema para un buen funcionamiento (MTTR)

La disponibilidad involucra medias poblacionales del tiempo entre fallas y el tiempo de reparación, estas se deben estimar ya que generalmente no se conocen las medias poblacionales:

$$(1.2) \quad \hat{A} = \frac{\hat{\mu}_{Tf}}{\hat{\mu}_{Tf} + \hat{\mu}_{Tr}}$$

donde:

\hat{A} = estimación de la disponibilidad

$\hat{\mu}_{Tf}$ = Estimación de la media de los tiempos entre fallas (MTBF) para una muestra.

$\hat{\mu}_{Tr}$ = Estimación de la media de los tiempos en restaurar el sistema para un buen funcionamiento (MTTR) para una muestra.

La calidad de la estimación de la disponibilidad inherente es dudosa si los tamaños de muestras son pequeños. Una solución a este problema es el uso de intervalos de confianza con un nivel de confianza dado. Sin embargo, los límites de confianza de la disponibilidad inherente no son fáciles de calcular [Wang y Kececioglu, (2000)]. En realidad, ninguna aproximación adecuada del cálculo del intervalo de confianza para la disponibilidad inherente, está disponible, basándose en datos de la muestra. Los límites

de confianza exactos sobre la disponibilidad inherente están dados sólo cuando los tiempos entre fallas y los tiempos de restauración se distribuyen exponencialmente. Basándose en suposiciones razonables, se han elaborado métodos no paramétricos para determinar la aproximación de los límites de confianza para la disponibilidad inherente de datos que son propuestos donde los tiempos entre fallas y los tiempos de reparación no se distribuyen exponencialmente.

Wang y Kececioglu (2000) han creado un método en el cual presentan por medio de una aproximación normal el cálculo del intervalo de confianza para la disponibilidad inherente, cuando los tiempos entre fallas y los tiempos de reparación no se distribuyen exponencialmente. Sin embargo, Wang y Kececioglu (2000) no realizaron un estudio que muestre el desempeño del método propuesto.

Efron desarrolló un método de estimación no paramétrico llamado Bootstrap, el cual requiere de un gran apoyo computacional. Este método representa una alternativa de solución para acercarse al problema de la precisión de las estimaciones a través de la construcción de Intervalos de confianza.

El método Bootstrap se diferencia del enfoque tradicional paramétrico en el que emplea un gran número de cálculos repetitivos para estimar la forma de la distribución muestral del estadístico, en lugar de fuertes afirmaciones distribucionales y fórmulas analíticas. Esto permite al investigador hacer

inferencias en casos en donde tales soluciones analíticas no son viables. Bootstrap es una aproximación al uso de la estadística para hacer inferencias sobre los parámetros poblacionales. El Bootstrap puede considerarse como un tipo especial de simulación denominada simulación basada en los datos [Efron y Tibshirani, (1993)].

Aunque existen varios tipos de metodología Bootstrap, en este trabajo solo se utilizará Bootstrap estándar, por ser el método más intuitivo [Efron, (1982 y 1987)].

Este proyecto de tesis se basa en gran parte en el trabajo de investigación realizado por Wang y Kececioglu (2000), quienes muestran un método utilizado para calcular la disponibilidad por medio de la aproximación de intervalos de confianza cuando los tiempos entre fallas y los tiempos de reparación no se distribuyen exponencialmente. Dicho método se comparará con el método Bootstrap estándar con el fin de probar la consistencia de resultados en ambos métodos.

1.1 Objetivo

El objetivo de esta tesis está constituido por dos grandes secciones: la primera de ellas es aplicar el método no paramétrico Bootstrap estándar para estimar la disponibilidad por Intervalos de confianza del 90% y 95%; la segunda parte es realizar un estudio de desempeño con el fin de probar la

consistencia de resultados entre el método no paramétrico Bootstrap estándar y el método utilizado por Wang y Kececioglu (2000).

1.2 Metodología

Para la realización de este trabajo de tesis se siguieron los pasos:

1. Investigación bibliográfica
2. Investigación bibliográfica de la técnica no paramétrica Bootstrap.
3. Investigación bibliográfica de los métodos estadísticos empleados para la validación de la simulación utilizada.
4. Diseño e implementación del algoritmo que genera muestras provenientes de las distribuciones empleadas en este trabajo de tesis y estima los porcentajes de cobertura de los intervalos de confianza para el 90% y 95% tanto para el método Wang-Kececioglu como para el método Bootstrap.
5. Ejecución del algoritmo. Para la simulación empleada se trabajó con una computadora PC y los códigos de los algoritmos fueron creados en el lenguaje de programación ANSI C [Gottfried (1997), Law y Kelton (1991)].
6. Validación de la simulación realizada.
7. Análisis de los resultados obtenidos.
8. Conclusiones y propuestas para trabajos a futuro.

1.3 Estructura de la tesis

En el capítulo 2 se introducen algunas definiciones de la disponibilidad de tal forma que el lector pueda entender más fácilmente el contenido de esta tesis.

En el capítulo 3 se definen algunos conceptos básicos de la metodología Bootstrap de tal forma que el lector pueda familiarizarse rápidamente con estos y así entender más fácilmente el resto de los capítulos.

En el capítulo 4 se detalla el análisis realizado para la estimación de los porcentajes de cobertura a partir de los límites de confianza del 90% y 95%.

En este mismo capítulo se lleva a cabo la validación de la simulación empleada para estimar los porcentajes de cobertura.

En el capítulo 5 se plantean las conclusiones de la tesis y los trabajos a futuro que pueden complementar esta investigación.

Finalmente en los Anexos se muestra el código en lenguaje C, empleado para llevar a cabo la simulación, y los resultados obtenidos de la misma.

Capítulo 2

Antecedentes

En este capítulo se introducen algunas definiciones de la disponibilidad de tal forma que el lector pueda entender más fácilmente el contenido de esta tesis. También se incluyen ejemplos relacionados con la estimación de intervalos de confianza.

2.1 Disponibilidad

La productividad de una planta es medida por la combinación de índices que son influenciados por la magnitud, frecuencia y duración de una falla. El conocido análisis de invariabilidad incluye costo-disponibilidad y costo-beneficio [Ramakumar, 1993]. El análisis de disponibilidad es una metodología que puede auxiliar a Ingenieros en la productividad de una planta, varias soluciones recomendadas para la productividad son basadas en la planeación y diseño de la operación, incluyendo el mantenimiento. De esta manera, la disponibilidad, confiabilidad, mantenimiento, costo de falla y el costo de duración del ciclo de un sistema, son indicadores que se pueden utilizar para evaluar el estado actual de la planta y encontrar áreas de oportunidad para mejorar la productividad.

Para caracterizar la confiabilidad de un sistema se requiere medir la disponibilidad, el concepto de disponibilidad es muy importante para el análisis de sistemas reparables, por ejemplo, sistemas de manufactura, vehículos de transporte, sistemas de protección contra incendios, etc.

La disponibilidad indica los porcentajes esperados de tiempo en que los productos van a operar satisfactoriamente. Es definida por:

$$(2.1) \quad A = \frac{\mu_{Tf}}{\mu_{Tf} + \mu_{Tr}}$$

donde:

μ_{Tf} = media de los tiempos entre fallas (MTBF)

μ_{Tr} = media de los tiempos en restaurar el sistema para un buen funcionamiento (MTTR)

La media del tiempo que se tardan en restaurar un sistema para un buen funcionamiento (MTTR), es una medida de mantenimiento. La media del tiempo de falla (MTTF) y la media del tiempo entre falla (MTBF) son medidas de confiabilidad. La media de tiempos entre fallas (MTBF) es el esperado o valor medio de una variable aleatoria llamada "tiempos entre fallas", es decir, el tiempo que el sistema trabaja correctamente hasta que falla. Si el tiempo de reparación (MTTR) es muy corto comparado con el tiempo entre fallas (MTBF), entonces la media del tiempo entre fallas (MTBF) es casi la misma

que la media de tiempo de falla (MTTF) y los dos términos son usados intercambiabilmente [Ramakumar, (1993)].

La disponibilidad inherente nos indica el porcentaje de tiempo que se espera que los sistemas funcionen satisfactoriamente. En realidad es muy raro conocer la verdadera disponibilidad inherente. Usualmente se estima recolectando datos de un sistema en operación, tomando el tiempo entre cada falla así como el tiempo que se tarda en reparación. La estimación puntual de la disponibilidad esta dada por:

$$(2.2) \quad \hat{A} = \frac{\hat{\mu}_{Tf}}{\hat{\mu}_{Tf} + \hat{\mu}_{Tr}}$$

donde:

\hat{A} = estimación de la disponibilidad

$\hat{\mu}_{Tf}$ = Estimación de la media de los tiempos entre fallas (MTBF) para una muestra.

$\hat{\mu}_{Tr}$ = Estimación de la media de los tiempos en restaurar el sistema para un buen funcionamiento (MTTR) para una muestra.

Suponiendo que n fallas son observadas, definiendo los tiempos entre fallas Tf_1, Tf_2, \dots, Tf_n y los tiempos de restauración definidos Tr_1, Tr_2, \dots, Tr_n . La estimación media de los tiempos entre fallas MTBF es dada por:

$$(2.3) \quad \hat{\mu}_{Tf} = \frac{1}{n}(Tf_1 + Tf_2 + \dots + Tf_n) = \frac{1}{n} \sum_{i=1}^n Tf_i$$

y la estimación media de los tiempos en reparación MTTR es dada por:

$$(2.4) \quad \hat{\mu}_{Tr} = \frac{1}{n}(Tr_1 + Tr_2 + \dots + Tr_n) = \frac{1}{n} \sum_{i=1}^n Tr_i$$

Los puntos estimados son funciones de los datos de la muestra y del tamaño de la muestra. Diferentes muestras producen diferentes estimaciones [Wang y Kececioglu, (2000)]. Si las estimaciones son basadas en una falla y en una reparación solamente, esto podría ser de riesgo. Es más confiable si se tienen más datos (más fallas y reparaciones), pero ¿Qué tan buena estimación de la disponibilidad inherente se puede tener?, Se sabe que la disponibilidad \hat{A} es un estimador de la disponibilidad de la población A [Ireson, (1966)].

La estimación del Intervalo de confianza nos indica el intervalo de valores dentro del cual es posible se encuentre el parámetro de población, en este caso sería A , la disponibilidad.

2.2 Intervalo de confianza para el caso Exponencial-Exponencial

Se desea mostrar el intervalo de confianza para la disponibilidad cuando los tiempos entre fallas T_f y los tiempos de reparación T_r siguen una distribución exponencial en ambos casos.

Entonces, tenemos que los tiempos entre fallas tienen distribución Exponencial(β), con $\beta = \mu_{T_f}$ y los tiempos de reparación Exponencial(β), con $\beta = \mu_{T_r}$. La suma de variables exponenciales nos da como resultado la distribución Gamma (α, β), un caso especial de la distribución Gamma es la distribución Ji-cuadradas con $\alpha = p/2$, $\beta = 2$ y con $2n$ grados de libertad [Casella y Berger, (1990)].

En el artículo realizado por Wang y Kececioglu (2000), se hace referencia a Kececioglu (1993). Tenemos:

$$(2.5) \quad \frac{2n\hat{\mu}_{T_f}}{\mu_{T_f}} \sim \chi_{2n}^2 \quad \text{y}$$

$$(2.6) \quad \frac{2n\hat{\mu}_{T_r}}{\mu_{T_r}} \sim \chi_{2n}^2$$

Se sabe que los tiempos entre fallas T_f y los tiempos en reparación T_r son independientes, la división de variables que se distribuyen Ji-cuadrada nos

da como resultado la distribución F con $2n$, $2n$ grados de libertad, [Casella y Berger, (1990)].

Tenemos:

$$(2.7) \quad \frac{\left[\frac{2n\hat{\mu}_{Tf}}{\mu_{Tf}} \right] / 2n}{\left[\frac{2n\hat{\mu}_{Tr}}{\mu_{Tr}} \right] / 2n} \sim F_{2n,2n}$$

$$(2.8) \quad \frac{\mu_{Tr} / \mu_{Tf}}{\hat{\mu}_{Tr} / \hat{\mu}_{Tf}} \sim F_{2n,2n}$$

Otra forma de expresar la disponibilidad es:

$$(2.9) \quad A = \frac{1}{1 + \frac{\mu_{Tr}}{\mu_{Tf}}}$$

Entonces:

$$(2.10) \quad \frac{\mu_{Tr}}{\mu_{Tf}} = \frac{1}{A} - 1$$

El estimado es:

$$(2.11) \quad \frac{\hat{\mu}_{Tr}}{\hat{\mu}_{Tf}} = \frac{1}{\hat{A}} - 1$$

Sustituimos las ecuaciones (2.10) y (2.11) en la ecuación (2.8), tenemos:

$$(2.12) \quad \frac{\frac{1}{A} - 1}{\frac{1}{\hat{A}} - 1} \sim F_{2n,2n}$$

Por lo tanto:

$$(2.13) \quad P \left(F_{1-\alpha/2, 2n, 2n} \leq \frac{\frac{1}{\hat{A}} - 1}{\frac{1}{\hat{A}}} \leq F_{\alpha/2, 2n, 2n} \right) = 1 - \alpha$$

Los Intervalos de confianza para la disponibilidad, cuando los tiempos entre fallas Tf y los tiempos de reparación Tr siguen una distribución exponencial satisfacen:

$$(2.14) \quad P \left(\frac{1}{1 + \left(\frac{1}{\hat{A}} - 1 \right) F_{\alpha/2, 2n, 2n}} \leq A \leq \frac{1}{1 + \left(\frac{1}{\hat{A}} - 1 \right) F_{1-\alpha/2, 2n, 2n}} \right) = 1 - \alpha$$

Generalmente, en muchos procesos de la vida real, los tiempos entre fallas y los tiempos de reparación no siempre se distribuyen exponencialmente. Por ejemplo, la distribución log-normal es usada preferentemente para los tiempos de reparación [Wang y Kececioglu (2000)]. Resulta entonces conveniente estimar intervalos de confianza para disponibilidad inherente cuando los tiempos entre fallas y los tiempos de reparación no se distribuyen exponencialmente.

2.2.1 Ejemplo del Intervalo de Confianza para el caso Exponencial-Exponencial

La siguiente tabla muestra un ciclo de 20 fallas y reparaciones consecutivamente. Se muestran los tiempos ficticios en horas de los tiempos

entre fallas y los tiempos de reparación de una máquina que está en constante operación. Se quiere obtener el intervalo de confianza de la disponibilidad con un nivel de significancia de $\alpha = 0.01$.

<i>Tf ~ Exp</i>	<i>Tr ~ Exp</i>
Tiempos entre fallas	Tiempos de reparación
0.5615	2.53826
1.2368	2.02305
3.2837	3.28978
6.0425	0.01484
0.4253	0.04893
10.1518	2.13802
1.5782	5.21748
4.2118	1.52494
6.1846	2.66602
7.0074	0.60277
14.1258	0.40751
5.1933	2.22821
0.2941	2.5672
5.4186	2.04218
9.8703	0.84993
0.0001	0.76247
7.8618	2.19515
8.8578	0.53836
0.3979	2.37908
15.2108	1.47882

La estimación media de los tiempos entre fallas MTBF está dada por:

$$\hat{\mu}_{Tf} = \frac{1}{n}(Tf_1 + Tf_2 + \dots + Tf_n) = \frac{1}{n} \sum_{i=1}^n Tf_i \quad \hat{\mu}_{Tf} = 5.39$$

La estimación media de los tiempos en reparación MTTR está dada por:

$$\hat{\mu}_{Tr} = \frac{1}{n}(Tr_1 + Tr_2 + \dots + Tr_n) = \frac{1}{n} \sum_{i=1}^n Tr_i \quad \hat{\mu}_{Tr} = 1.77$$

La estimación puntual de la disponibilidad inherente es:

$$\hat{A} = \frac{\hat{\mu}_{Tf}}{\hat{\mu}_{Tf} + \hat{\mu}_{Tr}} = \frac{5.39}{5.39 + 1.77} = 0.7529$$

614124

Los Intervalos de confianza para la disponibilidad, cuando los tiempos entre fallas T_f y los tiempos de reparación T_r siguen una distribución exponencial satisfacen:

$$P \left(\frac{1}{1 + \left(\frac{1}{\hat{A}} - 1 \right) F_{\alpha/2, 2n, 2n}} \leq A \leq \frac{1}{1 + \left(\frac{1}{\hat{A}} - 1 \right) F_{1-\alpha/2, 2n, 2n}} \right) = 1 - \alpha :$$

Con $\alpha = 0.1$, buscamos el valor del estadístico F en las tablas correspondientes con $\alpha/2 = 0.05$ y $2n$ grados de libertad, $F_{0.05, 40, 40} = 1.69$ y $F_{0.95, 40, 40} = 0.69$. Sustituimos los valores correspondientes:

$$P \left(\frac{1}{1 + \left(\frac{1}{0.7529} - 1 \right) 1.69} \leq A \leq \frac{1}{1 + \left(\frac{1}{0.7529} - 1 \right) 0.69} \right) = 0.90$$

$$0.6432 \leq A \leq 0.815357$$

2.3 Método Wang-Kececioglu

En el año 2000, Wang y Kececioglu publican un artículo en el cual proponen un método para la aproximación de la disponibilidad por Intervalos de confianza.

Se considera la estimación puntual de la disponibilidad como:

$$(2.15) \quad \hat{A} = \frac{1}{1 + \frac{\hat{\mu}_{Tr}}{\hat{\mu}_{Tf}}}$$

Los tiempos entre fallas son usualmente mucho más grandes que los tiempos de reparación, entonces;

$$(2.16) \quad \frac{\hat{\mu}_{Tr}}{\hat{\mu}_{Tf}} \ll 1$$

Y la expresión de la estimación de la disponibilidad puede ser expresada por medio de una serie de Taylor:

$$(2.17) \quad \hat{A} = 1 - \frac{\hat{\mu}_{Tr}}{\hat{\mu}_{Tf}} + \left(\frac{\hat{\mu}_{Tr}}{\hat{\mu}_{Tf}}\right)^2 - \left(\frac{\hat{\mu}_{Tr}}{\hat{\mu}_{Tf}}\right)^3 + \left(\frac{\hat{\mu}_{Tr}}{\hat{\mu}_{Tf}}\right)^4 + \dots,$$

Lo cual significa que la estimación de disponibilidad como una variable arbitraria puede ser expresada como la suma de un número grande de variables arbitrarias. Por el teorema central del límite, la estimación de la disponibilidad tiende a distribuirse aproximadamente como una normal [Kececioglu (1995)].

Por lo tanto, una buena aproximación para la estimación de la disponibilidad [Wang y Kececioglu (2000)] es:

$$(2.18) \quad \hat{A} \sim N(\bar{A}, s_A)$$

donde:

\bar{A} = media de la estimación de la disponibilidad inherente

s_A = desviación estándar de la estimación de la disponibilidad inherente.

Método:

Los parámetros se aproximan a la distribución normal aplicando el método de momentos a la ecuación (2.1). En el artículo realizado por Wang y Kececioglu (2000) se menciona como referencia, Kececioglu (1995):

$$(2.19) \quad \bar{A} = \frac{\mu_{Tf}}{\mu_{Tf} + \mu_{Tr}} + \frac{1}{2} \left(\frac{\partial^2 \hat{A}}{\partial \hat{\mu}_{Tf}^2} \right) \sigma_{\mu_{Tf}}^2 + \frac{1}{2} \left(\frac{\partial^2 \hat{A}}{\partial \hat{\mu}_{Tr}^2} \right) \sigma_{\mu_{Tr}}^2$$

$$(2.20) \quad \bar{A} = \frac{\mu_{Tf}}{\mu_{Tf} + \mu_{Tr}} + \frac{\mu_{Tf} s_{Tr}^2 - \mu_{Tr} s_{Tf}^2}{n(\mu_{Tf} + \mu_{Tr})^3} = A + \Delta A$$

y

$$(2.21)$$

$$s_A^2 = \left(\frac{\partial \hat{A}}{\partial \hat{\mu}_{Tf}} \right)^2 \sigma_{\mu_{Tf}}^2 + \left(\frac{\partial \hat{A}}{\partial \hat{\mu}_{Tr}} \right)^2 \sigma_{\mu_{Tr}}^2 + \left(\frac{\partial \hat{A}}{\partial \hat{\mu}_{Tf}} \right) \left(\frac{\partial^2 \hat{A}}{\partial \hat{\mu}_{Tr}^2} \right) m_{3, \mu_{Tf}} + \left(\frac{\partial \hat{A}}{\partial \hat{\mu}_{Tr}} \right) \left(\frac{\partial^2 \hat{A}}{\partial \hat{\mu}_{Tf}^2} \right) m_{3, \mu_{Tr}}$$

o

$$(2.22) \quad s_A^2 = \frac{\mu_{Tr}^2 s_{Tf}^2 + \mu_{Tf}^2 s_{Tr}^2}{n(\mu_{Tf} + \mu_{Tr})^4} - \frac{2\mu_{Tr}^2 m_{3, \mu_{Tf}} + 2\mu_{Tf}^2 m_{3, \mu_{Tr}}}{n^2(\mu_{Tf} + \mu_{Tr})^5}$$

donde:

s_{Tf} = desviación estándar de los tiempos entre fallas

s_{Tr} = desviación estándar de los tiempos de reparación

$m_{3, \mu_{Tf}}$ = sesgo de los tiempos entre fallas

$m_{3_{\mu_{Tr}}}$ = sesgo de los tiempos de reparación

Aproximación de Intervalos de Confianza [Wang y Kececioglu (2000)]:

Vemos que el primer término de la ecuación (2.20), es exactamente el verdadero valor de la disponibilidad. Esto se puede reescribir como:

$$(2.23) \quad \bar{A} = A + \Delta A$$

La probabilidad es:

$$(2.24) \quad P\left\{-z_{\alpha/2} < \frac{\hat{A} - (A + \Delta A)}{s_A} < z_{\alpha/2}\right\} = 1 - \alpha$$

Por lo tanto, un intervalo de confianza aproximado es:

$$(2.25) \quad P\left\{\hat{A} - \Delta A - s_A \cdot z_{\alpha/2} < A < \hat{A} - \Delta A + s_A \cdot z_{\alpha/2}\right\} = 1 - \alpha$$

2.3.1 Ejemplo del método Wang-Kececioglu

La siguiente tabla muestra un ciclo de 30 tiempos de fallas y reparaciones consecutivamente. Se muestran los tiempos ficticios en horas de los tiempos entre fallas y los tiempos de reparación de una máquina que se encuentra en constante operación. Los datos de los tiempos entre fallas siguen una distribución exponencial, los datos de los tiempos de reparación se

distribuyen Gamma. Se quiere obtener el intervalo de confianza de la disponibilidad con un nivel de significancia de $\alpha = 0.01$.

<i>Tf ~ Exp</i>	<i>Tr ~ Gamma</i>
Tiempos entre fallas	Tiempos de reparación
5.1466	0.45406
1.4267	0.79035
1.9063	0.22271
5.8714	0.75964
3.2487	0.42358
4.184	0.373
1.4915	0.80857
3.1393	1.5148
11.067	1.2527
1.7523	0.3309
2.9425	0.87979
3.3384	0.74544
6.4067	0.81335
1.7041	1.74935
3.2404	0.52959
2.7462	0.7421
2.1586	0.93392
3.4612	0.56862
7.8374	1.81511
5.7424	0.84176
2.5628	0.78484
0.3048	0.81037
3.3306	0.85871
6.1476	0.87898
0.721	1.64995
1.7797	1.19247
4.9228	0.99546
3.2228	0.51342
2.7276	1.95409
1.2079	1.25178

La estimación media de los tiempos entre fallas MTBF es dada por:

$$\hat{\mu}_{Tf} = \frac{1}{n}(Tf_1 + Tf_2 + \dots + Tf_n) = \frac{1}{n} \sum_{i=1}^n Tf_i \quad \hat{\mu}_{Tf} = 3.52$$

$$\hat{\sigma}_{Tf}^2 = 5.3717$$

La estimación de momentos para los tiempos entre fallas:

$$\hat{m}_{3_{\mu_{Tf}}} = \frac{1}{n} \sum_{i=1}^n (Tf_i - \hat{\mu}_{Tf})^3 = 15.7714$$

La estimación media de los tiempos en reparación MTTR es dada por:

$$\hat{\mu}_{Tr} = \frac{1}{n} (Tr_1 + Tr_2 + \dots + Tr_n) = \frac{1}{n} \sum_{i=1}^n Tr_i \quad \hat{\mu}_{Tr} = 0.92$$

$$\hat{\sigma}_{Tr}^2 = 0.1984$$

La estimación de momentos para los tiempos entre fallas:

$$\hat{m}_{3_{\mu_{Tr}}} = \frac{1}{n} \sum_{i=1}^n (Tr_i - \hat{\mu}_{Tr})^3 = 0.0653$$

La estimación puntual de la disponibilidad inherente es:

$$\hat{A} = \frac{\hat{\mu}_{Tf}}{\hat{\mu}_{Tf} + \hat{\mu}_{Tr}} = \frac{3.52}{3.52 + 0.92} = 0.7927$$

Sustituimos la estimación de momentos en la ecuación (2.20), tenemos:

$$\Delta \hat{A} = -0.001551$$

Sustituimos los valores correspondientes en la ecuación (2.21):

$$\hat{s}_A = 0.0457$$

El intervalo de confianza para la disponibilidad es:

$$P\left\{\hat{A} - \Delta A - s_A \cdot z_{\alpha/2} < A < \hat{A} - \Delta A + s_A \cdot z_{\alpha/2}\right\} = 1 - \alpha$$

$$P(0.7927 + 0.001551 - 0.0457 \cdot 1.645 < A < 0.7927 + 0.001551 + 0.0457 \cdot 1.645) = 0.90$$

$$(0.71901 < A < 0.8694)$$

Capítulo 3

Método Bootstrap

En este capítulo se incluyen conceptos relacionados con la metodología estadística no paramétrica Bootstrap.

3.1 Método Bootstrap

En la actualidad se han desarrollado una serie de métodos basados en cálculos intensivos por medio de simulaciones destinadas a obtener medidas de la precisión de las estimaciones (errores típicos, sesgos, intervalos de confianza). Entre tales métodos se encuentra el método Bootstrap, introducido por Efron en el año 1979.

El Bootstrap descansa en la analogía entre la muestra y la población de la cual la muestra es extraída. De acuerdo a Efron y Tibshirani (1986) dada una muestra con n observaciones el estimador no paramétrico de máxima verosimilitud de la distribución poblacional es la función de densidad de probabilidad que asigna una masa de probabilidad de $\frac{1}{n}$ a cada una de las observaciones. La idea central es que muchas veces puede ser mejor

extraer conclusiones sobre las características de la población estrictamente a partir de la muestra que se maneja, que haciendo suposiciones probablemente poco realistas sobre la población. El *Bootstrapping* implica remuestreo (*resamplign*) de los datos obtenidos en una muestra, con reemplazo, muchas veces para generar una estimación empírica de la distribución muestral de un estadístico.

Algunos autores fijan la utilidad del método Bootstrap a tres niveles, [Lunneborg (1987)]:

1. Valorar el sesgo y el error muestral de un estadístico calculado a partir de una muestra.
2. Establecer un intervalo de confianza para un parámetro estimado.
3. Realizar una prueba de hipótesis respecto a uno o más parámetros poblacionales.

3.2 Procedimiento del método Bootstrap

La idea básica del Bootstrap es tratar a la muestra como si fuera la población y aplicar el método de muestreo Monte Carlo para generar una estimación empírica de la distribución muestral del estadístico. La estimación Monte Carlo requiere del conocimiento de la población. Típicamente tenemos sólo una muestra extraída de esa población, por lo cual necesitamos, antes que nada, inferir θ a partir de $\hat{\theta}$.

En el método Bootstrap se trata a la muestra como si fuera la población y se realizan procedimientos del método Monte Carlo sobre la muestra. Esto se hace extrayendo un gran número de muestras de tamaño n – las cuales denominaremos como “remuestras”- de la muestra original aleatoriamente y con reemplazo. Así, aunque cada remuestra tendrá el mismo número de elementos que la muestra original, mediante el remuestreo con reemplazo cada muestra podrá tener algunos de los datos originales representados en ella más de una vez, y algunos que no aparecieron. Por lo tanto, cada una de estas remuestras probablemente serán aleatoriamente diferentes de la muestra original. Y como los elementos en estas remuestras varían levemente, un estadístico $\hat{\theta}^*$, calculado a partir de una de esas remuestras probablemente tomará un valor ligeramente diferente de los otros $\hat{\theta}^*$ y del $\hat{\theta}$ original. La afirmación fundamental del Bootstrap es que una distribución de frecuencias relativas de esos $\hat{\theta}^*$ calculado a partir de las remuestras es una estimación de la distribución muestral de $\hat{\theta}$ [Mooney y Duval (1993)].

Más formalmente, los pasos básicos en la estimación Bootstrap son los siguientes [Efron, (1979); Hinckley, (1988); Efron y Tibshirani, (1993)]:

1. Construir una distribución de probabilidad empírica $\hat{F}(x)$, a partir de las muestras asignando una probabilidad de $\frac{1}{n}$ a cada punto, x_1, x_2, \dots, x_n .

Esta es la función de distribución empírica $FDE(x)$, la cual es el

estimador no paramétrico de máxima verosimilitud de la función de distribución de la población, $F(x)$.

2. A partir de la FDE, $\hat{F}(x)$ se extrae una muestra aleatoria simple de tamaño n con reemplazo. La cual es una “remuestra”, x_b^* .
3. Se calcula el estadístico de interés $\hat{\theta}$, a partir de esa remuestra, dando $\hat{\theta}_b^*$.
4. Se repiten los pasos 2 y 3 B veces, donde B es un número grande. La magnitud de B en la práctica depende de las pruebas que se van a aplicar a los datos. En general, B debería ser de entre 50 a 200 para estimar el error típico de $\hat{\theta}$, y al menos de 1000 para estimar intervalos de confianza alrededor de $\hat{\theta}$ [Efron y Tibshirani, (1986, 1993)].
5. Construir una distribución de probabilidad a partir de los B $\hat{\theta}_b^*$ asignando una probabilidad de $\frac{1}{B}$ a cada punto, $\hat{\theta}_1^*, \hat{\theta}_2^*, \dots, \hat{\theta}_B^*$. Esta distribución es la estimación Bootstrap de la distribución muestral de $\hat{\theta}$, $\hat{F}^*(\hat{\theta}^*)$. Esta distribución puede usarse para hacer inferencias sobre θ .
6. El estimador Bootstrap del parámetro θ se define como:

$$(3.1) \quad \hat{\theta}^* = \frac{\sum_{i=1}^B \hat{\theta}_i}{B}$$

es decir, como la media de los valores de los estadísticos calculados en las B remuestras Bootstrap. Y la estimación de la desviación estándar del parámetro es:

$$(3.2) \quad S_{\hat{\theta}} = \sqrt{\frac{\sum_{b=1}^B (\hat{\theta}_b - \hat{\theta}^*)^2}{B-1}}$$

Por lo tanto, el método Bootstrap es una técnica fácil para implantar en un sistema computacional que realice las simulaciones del método. Una ventaja de este método es que no requiere una expresión analítica para el estimador. Se pueden realizar simulaciones Bootstrap muy fácilmente una vez que se programen las réplicas, por medio de un algoritmo, entonces la computadora hace todo el trabajo.

En esta tesis se empleará el método Bootstrap estándar en la estimación de intervalos de confianza para la disponibilidad.

3.3 Empleo del método Bootstrap estándar en la estimación de intervalos de confianza para la disponibilidad

Este método es similar al procedimiento paramétrico de construcción de intervalos de confianza. A partir de una muestra de tamaño n se pueden obtener n^n remuestras con reemplazo Bootstrap de las cuales es posible calcular n^n valores de la disponibilidad estimada.

Para el caso de intervalos de confianza, se emplea un total de $B=1000$ muestras bootstrap que permiten realizar las estimaciones bootstrap estándar de los intervalos de confianza para la disponibilidad al $(1-\alpha)100\%$, de la siguiente forma:

$$(3.3) \quad \hat{A} \pm Z_{\alpha/2} S_{\hat{\theta}}$$

donde:

$$(3.4) \quad S_{\hat{\theta}} = \sqrt{\frac{\sum_{b=1}^B (\hat{\theta}_b - \hat{\theta}^*)^2}{B-1}}$$

donde:

$$(3.5) \quad \hat{\theta}^* = \frac{\sum_{i=1}^B \hat{\theta}_i}{B}$$

donde:

\hat{A} , es el estimador de la disponibilidad obtenida a partir de la muestra original.

$\hat{\theta}_i^*$, es la i -ésima estimación bootstrap de la disponibilidad.

$\hat{\theta}^*$, es la media aritmética de los $\hat{\theta}_i^*$.

$S_{\hat{\theta}}$, es el estimador de la desviación estándar de \hat{A} .

$Z_{\alpha/2}$, es el cuantil $\alpha/2$ de la distribución normal estándar.

Si alguna disponibilidad (A) real cae dentro de un respectivo límite de confianza, ello deberá ser reflejado como un porcentaje de cobertura.

3.4 Representación de la simulación Bootstrap

Este método forma parte de una clase de métodos que utilizan muestras obtenidas a partir de un conjunto de datos originales y por ellos son llamados métodos de remuestreo. Enseguida, se muestra la metodología a seguir para la aplicación del método Bootstrap estándar utilizado para calcular intervalos de confianza para la disponibilidad:

	Tiempos entre fallas	Tiempos en reparación	
Muestras Originales	$Tf=(x_1, x_2, \dots, x_n) \rightarrow \bar{x}$	$Tr=(y_1, y_2, \dots, y_n) \rightarrow \bar{y}$	$\hat{A} = \frac{\bar{x}}{\bar{x} + \bar{y}}$
m.a.c.1 con reemplazo	$(x_1^{*1}, x_2^{*1}, \dots, x_n^{*1}) \rightarrow \bar{x}^{*1}$	$(y_1^{*1}, y_2^{*1}, \dots, y_n^{*1}) \rightarrow \bar{y}^{*1}$	$\hat{A}^{*1} = \frac{\bar{x}^{*1}}{\bar{x}^{*1} + \bar{y}^{*1}}$
m.a.c.2 con reemplazo	$(x_1^{*2}, x_2^{*2}, \dots, x_n^{*2}) \rightarrow \bar{x}^{*2}$	$(y_1^{*2}, y_2^{*2}, \dots, y_n^{*2}) \rightarrow \bar{y}^{*2}$	$\hat{A}^{*2} = \frac{\bar{x}^{*2}}{\bar{x}^{*2} + \bar{y}^{*2}}$
⋮	⋮	⋮	⋮
m.a.c. B con reemplazo	$(x_1^{*B}, x_2^{*B}, \dots, x_n^{*B}) \rightarrow \bar{x}^{*B}$	$(y_1^{*B}, y_2^{*B}, \dots, y_n^{*B}) \rightarrow \bar{y}^{*B}$	$\hat{A}^{*B} = \frac{\bar{x}^{*B}}{\bar{x}^{*B} + \bar{y}^{*B}}$
			↓
			$S_{\hat{A}}$

Se obtienen 1000 $\rightarrow IC_A = \hat{A} \pm Z_{\alpha/2} S_{\hat{A}}$

*m.a.c. ; muestra aleatoria continua

Figura 3.1 Proceso para estimar el intervalo de confianza del estadístico $\hat{\theta}$, mediante Bootstrap.

Para realizar la simulación antes mencionada se utilizó el lenguaje de programación C [Gottfried (1997)]. Los algoritmos empleados en esta simulación se encuentran en el Anexo D.

Capítulo 4

Estudio de desempeño

En este capítulo se describe, el diseño, implantación y validación del sistema computacional empleado para generar los porcentajes de cobertura de la disponibilidad para los métodos empleados.

4.1 Descripción de la simulación

Se desarrollo una serie de simulaciones para comparar el desempeño del método Bootstrap y el método propuesto por Wang y Kececioglu (2000). En las simulaciones se emplearon los siguientes valores de parámetros fijos reales para cada una de las distribuciones. Para tomar estos datos como valores de parámetros fijos reales, se tomó en cuenta los valores que proponen Wang y Kececioglu (2000) en un ejemplo de la estimación de intervalos:

Parámetros reales para las distribuciones del tiempo entre falla:

$$\mu_{Tf} = 55.1 \quad \sigma_{Tf}^2 = 2747.49$$

Parámetros reales para las distribuciones en el tiempo de reparación:

$$\mu_{Tr} = 2.49 \quad \sigma_{Tr}^2 = 7.7919$$

Y de esta forma medir la disponibilidad real teórica de este sistema.

Con el afán de probar la robustez del método Bootstrap se decidió utilizar distribuciones sesgadas y simétricas, se utilizaron las combinaciones de 6 distribuciones de probabilidad (normal, exponencial, t de student, log-normal, gamma, weibull). Para cada combinación se obtuvieron cuatro muestras originales de tamaño 20, 30, 50 y 100, y a partir de ellas se obtuvieron $B = 1000$ muestras Bootstrap. Se construyeron límites de confianza al 90% y 95%.

Los parámetros de las funciones de probabilidad con los cuales se llevó a cabo la ejecución de las simulaciones son lo siguientes:

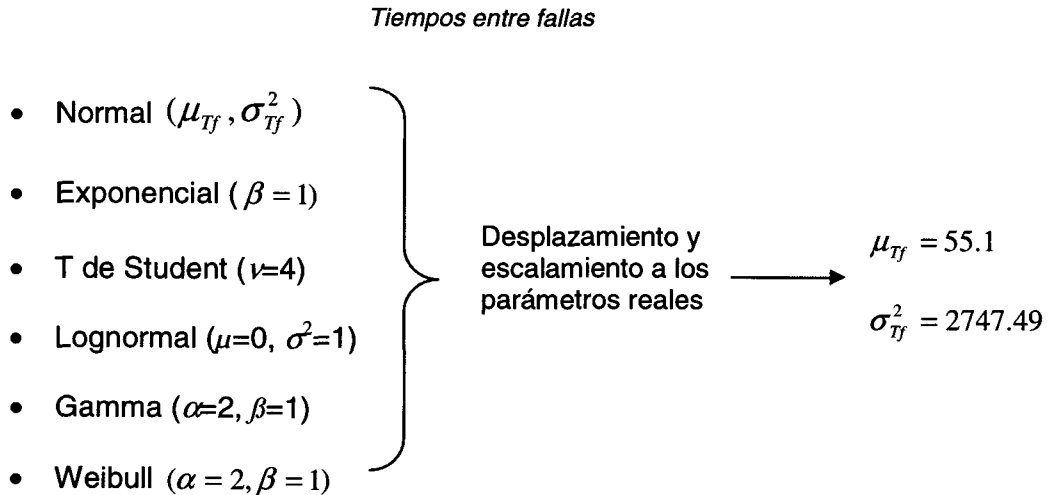


Figura 4.1 Distribuciones utilizadas para los tiempos entre fallas

Tiempos de reparación

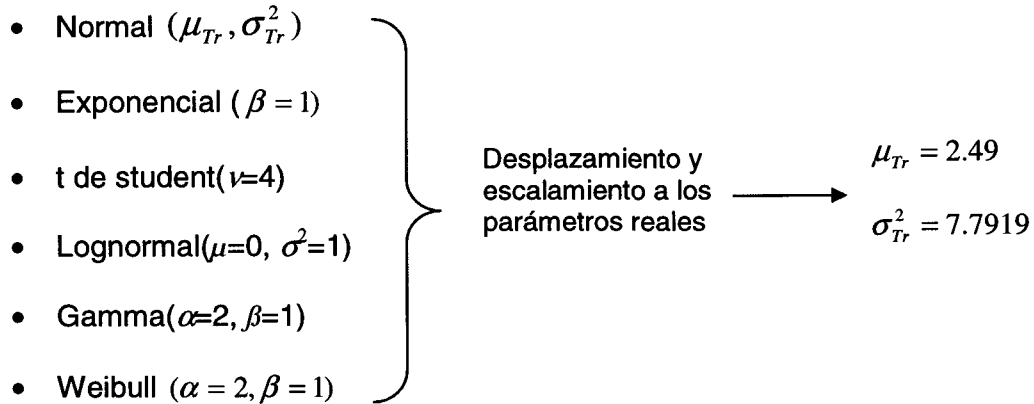
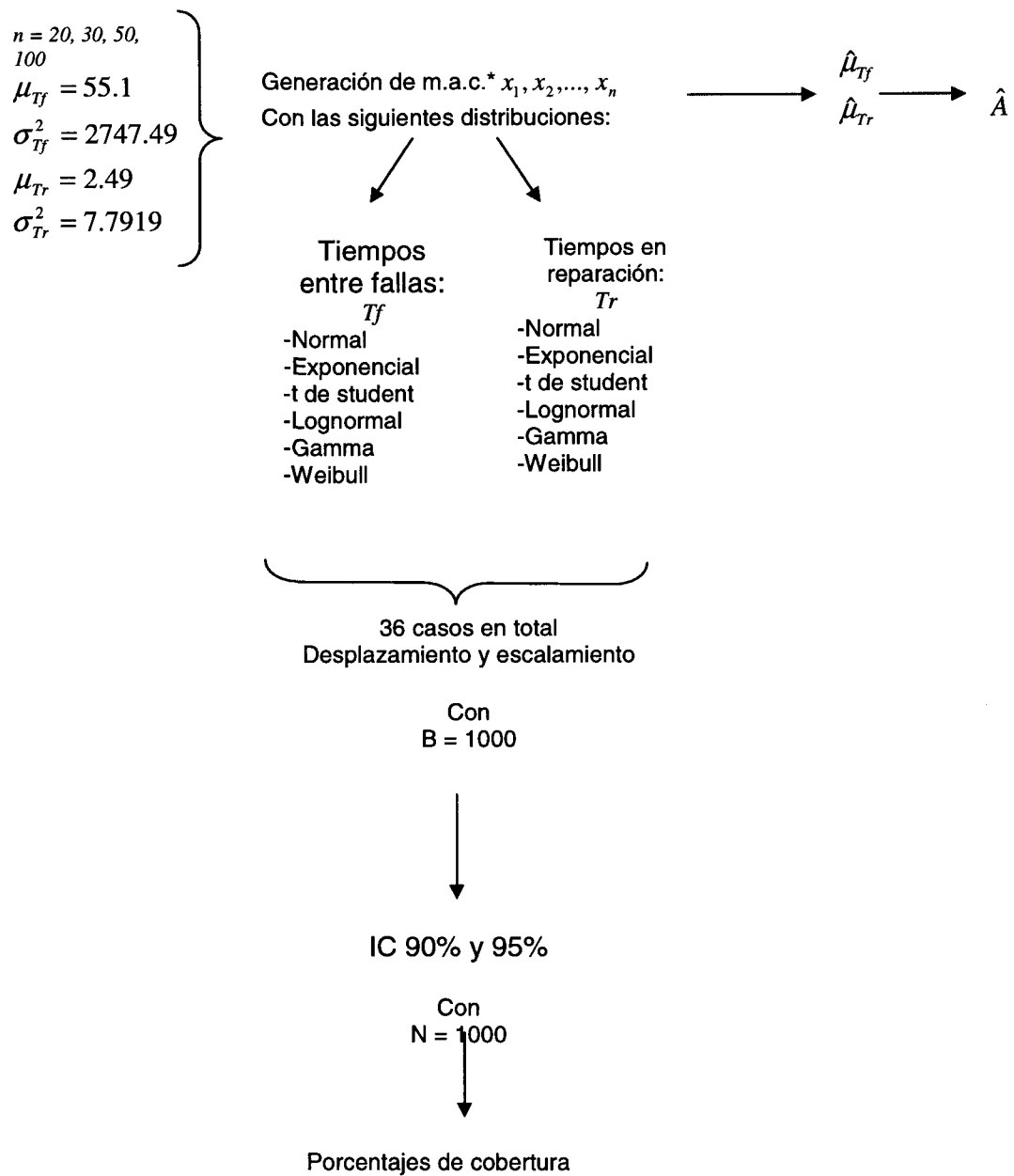


Figura 4.2 Distribuciones utilizadas para los tiempos de reparación

Se realizaron las combinaciones posibles con las 6 distribuciones, tanto para el tiempo de falla como para el tiempo de reparación, obteniendo intervalos de confianza del 90% y 95%.

Se le llama porcentaje de cobertura al el porcentaje de los intervalos de confianza que incluyen al valor real de la disponibilidad A . Este porcentaje de cobertura podría ser entonces comparado al valor esperado de 0.95 para el caso del intervalo de confianza del 95% o el valor de 0.90 para el caso de intervalo de confianza del 90%. Cada distribución fue desplazada y escalada para obtener las medias y las desviaciones estándar de tal forma que se puedan comparar con los valores de los parámetros reales.



* muestra aleatoria continua

Figura 4.3 Parámetros y distribuciones empleadas

Se diseñó e implementó un sistema computacional que realiza las simulaciones descritas anteriormente, teniendo parámetros fijos para cada una de las distribuciones y de esta forma medir la disponibilidad real teórica de este sistema. El proceso que se muestra a continuación se repitió $N=1000$ veces para cada una de las 6 distribuciones de interés.

4.2 Funciones de probabilidad de variables aleatorias continuas

Para llevar a cabo las simulaciones fue necesario generar variables aleatorias continuas. Las funciones de densidad de probabilidad utilizadas en este trabajo son, normal, exponencial, t de student, lognormal, gamma, y weibull. A continuación se muestran, las funciones, parámetros y los pseudocódigos de los algoritmos que generan las variables aleatorias de cada una de las funciones de probabilidad con los cuales se llevo a cabo la ejecución de las simulaciones:

- *Generación de variables aleatorias continuas con distribución Uniforme(0,1)*

No se describe un pseudocódigo del algoritmo para este caso particular. El algoritmo fue tomado tal y como aparece en la función denominada *ran2* de la del libro Law y Kelton (1991).

- *Generación de variables aleatorias continuas con distribución Normal (0,1)*

El algoritmo se conoce como método polar y genera variables aleatorias Normal (0,1). El pseudocódigo fue tomado del libro Law y Kelton (1991) y se muestra a continuación:

Hacer
 $U_1 = \text{Uniforme}(0,1)$
 $V_1 = 2U_1 - 1$
 $U_2 = \text{Uniforme}(0,1)$
 $V_2 = 2U_2 - 1$
 $W = V_1^2 + V_2^2$
Mientras (($W > 1$) ó ($W = 0$))
 $Y = \sqrt{\frac{-2\ln(W)}{W}}$
 $X_1 = V_1 Y$
 $X_2 = V_2 Y$

- *Generación de variables aleatorias continuas con distribución Normal(μ, σ^2)*

$Z = \text{Normal}(0,1)$
 $X = \mu + \sigma Z$

- Normal (μ, σ^2)

$$f(x, \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad -\infty < x < \infty, -\infty < \mu < \infty, \sigma > 0$$

$$E(x) = \mu \quad V(x) = \sigma^2$$

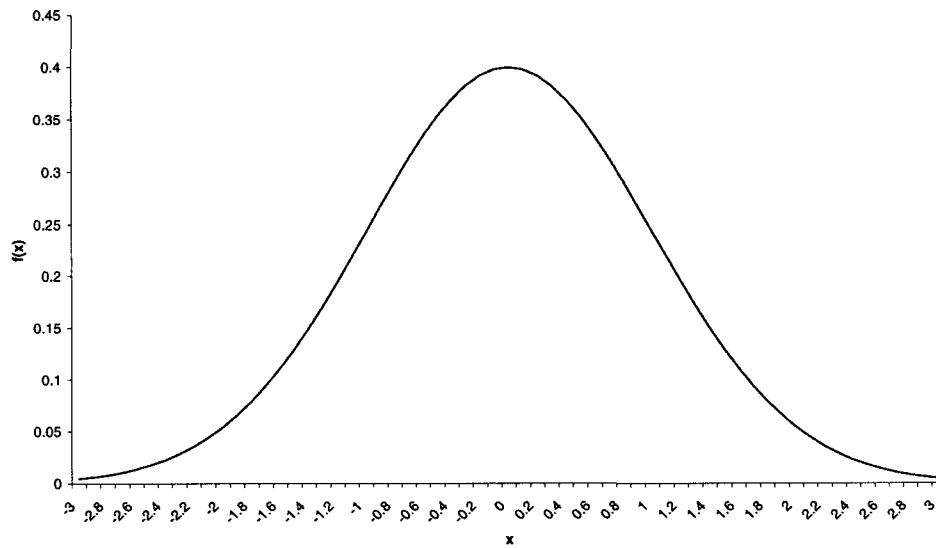


Figura 4.4 Distribución normal estándar o Normal($\mu=0$, $\sigma^2=1$)

- *Generación de variables aleatorias continuas con distribución Exponencial(β)*

El pseudocódigo de este algoritmo se obtuvo mediante el método de la transformación inversa [Law y Kelton (1991)] y se muestra a continuación:

$$U = \text{Uniforme}(0,1)$$

$$X = -\beta \ln(U)$$

- Exponencial (β)

$$f(x, \beta) = \frac{1}{\beta} e^{-\frac{x}{\beta}}, \quad 0 \leq x < \infty, \quad \beta > 0$$

$$E(x) = \beta \quad V(x) = \beta^2$$

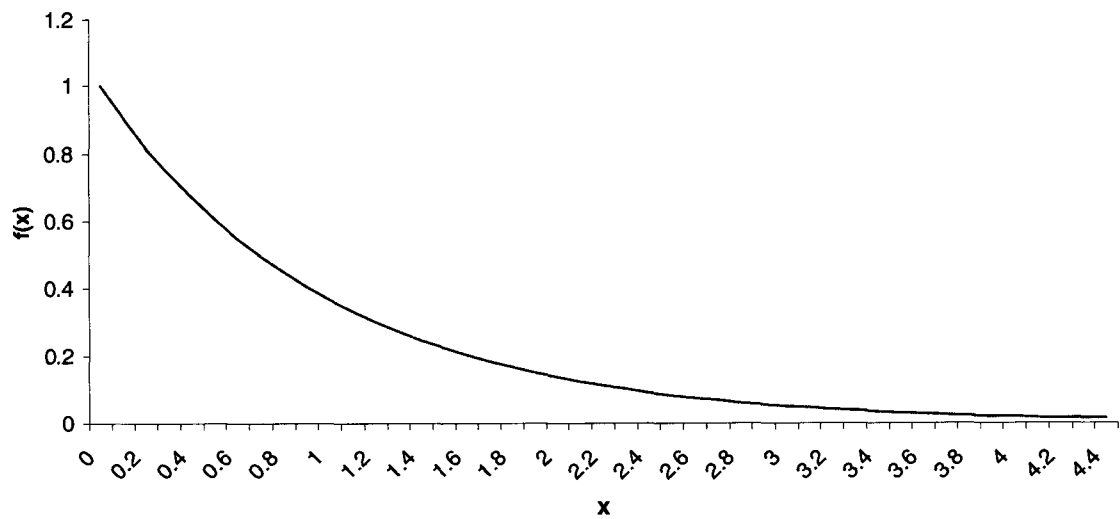


Figura 4.5 Distribución Exponencial ($\beta=1$)

- *Generación de variables aleatorias continuas con distribución t de student(ν)*

A partir de su definición, se pueden generar variables aleatorias con distribución t de student con ν grados de libertad [Casella y Berger (1990)] de la siguiente manera:

$$\begin{aligned}
 Z &= \text{Normal}(0,1) \\
 Y &= \text{Ji cuadrada}(\nu) \\
 X &= \frac{Z}{\sqrt{Y/\nu}}
 \end{aligned}$$

- T de Student (ν)

$$f(x, \mu, \sigma^2) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)} \frac{1}{\sqrt{\nu\pi}} \frac{1}{\left(1 + \frac{x^2}{\nu}\right)^{\frac{\nu+1}{2}}}, \quad -\infty < x < \infty, \nu \geq 1, \nu \in N$$

$$E(x) = 0 \quad V(x) = \frac{\nu}{\nu-2}$$

$$\nu > 1 \quad \nu > 2$$

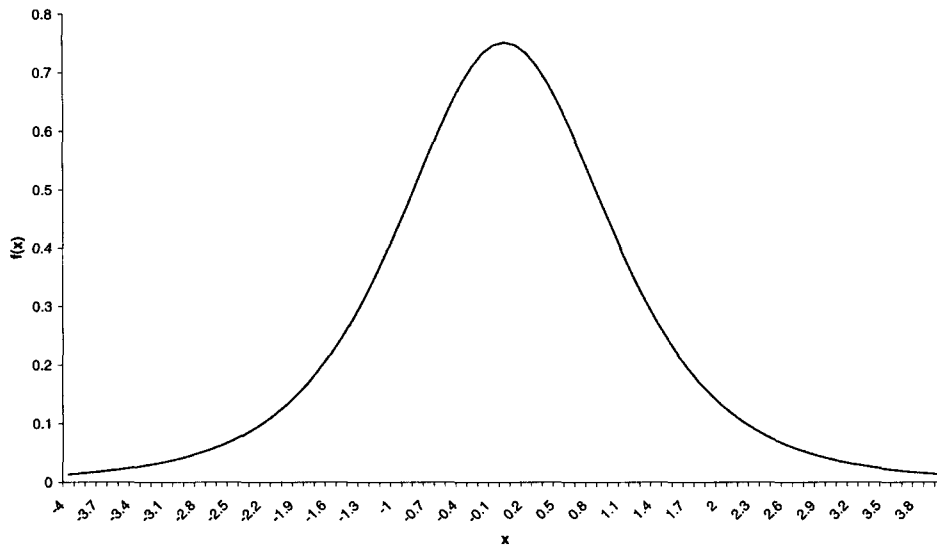


Figura 4.6 Distribución t de student($\nu=4$)

- *Generación de variables aleatorias continuas con distribución Lognormal(μ, σ^2)*

$$Y = \text{Normal}(\mu, \sigma^2)$$

$$X = e^Y$$

- Lognormal (μ, σ^2)

$$f(x, \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma}} \frac{e^{-\frac{[\ln(x)-\mu]^2}{2\sigma^2}}}{x}, \quad x > 0, -\infty < \mu < \infty, \sigma > 0$$

$$E(x) = e^{\mu + \frac{\sigma^2}{2}} \quad V(x) = e^{2\mu + \sigma^2} (e^{\sigma^2} - 1)$$

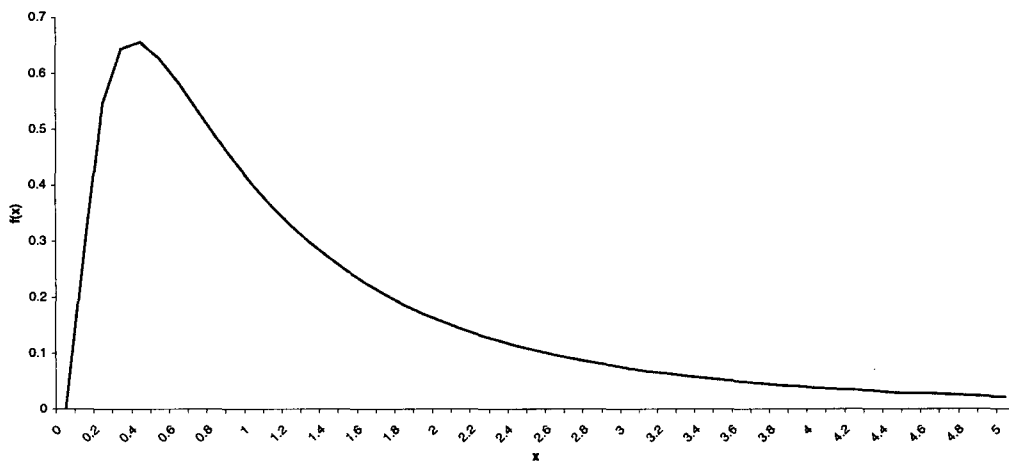


Figura 4.7 Distribución Lognormal($\mu=0, \sigma^2=1$)

- Generación de variables aleatorias continuas con distribución Gamma(α, β)

El pseudocódigo de este algoritmo aparece el libro de Law y Kelton (1991) y es el siguiente:

$$Y = \text{Gamma}(\alpha, 1)$$

$$X = \beta Y$$

Existen 3 casos para la obtención del valor de $Y = \text{Gamma}(\alpha, 1)$:

*** Caso $\alpha=1$**

$$Y = \text{Exponencial}(1)$$

*** Caso $\alpha < 1$**

$$b = \frac{e + \alpha}{e} = 1 + \frac{\alpha}{e}$$

Hacer

$$U_1 = \text{Uniforme}(0, 1)$$

$$U_2 = \text{Uniforme}(0, 1)$$

$$P = bU_1$$

Si ($P > 1$)

$$Y = -\ln\left(\frac{b-P}{\alpha}\right)$$

$$\text{comp} = Y^{\alpha-1}$$

De lo contrario

$$Y = P^{1/\alpha}$$

$$\text{comp} = e^{-Y}$$

Mientras ($U_2 > \text{comp}$)

*** Caso $\alpha > 1$**

$$a = \frac{1}{\sqrt{2\alpha - 1}}$$

$$b = \alpha \ln(4)$$

$$q = \alpha + \frac{1}{a}$$

$$\theta = 4.5$$

$$d = 1 + \ln(\theta)$$

Hacer

$$U_1 = \text{Uniforme}(0, 1)$$

$$U_2 = \text{Uniforme}(0, 1)$$

$$V = a \cdot \ln\left(\frac{U_1}{1-U_1}\right)$$

$$Y = \alpha e^V$$

$$Z = U_1^2 U_2$$

$$W = b + qV - Y$$

$$\text{comp} = W + d - \theta Z$$

Si ($\text{comp} < 0$)

$$\text{comp} = W - \ln(Z)$$

Mientras ($\text{comp} < 0$)

- Gamma (α, β)

$$f(x, \alpha, \beta) = \frac{1}{\Gamma(\alpha)\beta^\alpha} x^{\alpha-1} e^{-\frac{x}{\beta}}, \quad x \geq 0, \alpha > 0, \beta > 0$$

$$E(x) = \alpha\beta \quad V(x) = \alpha\beta^2$$

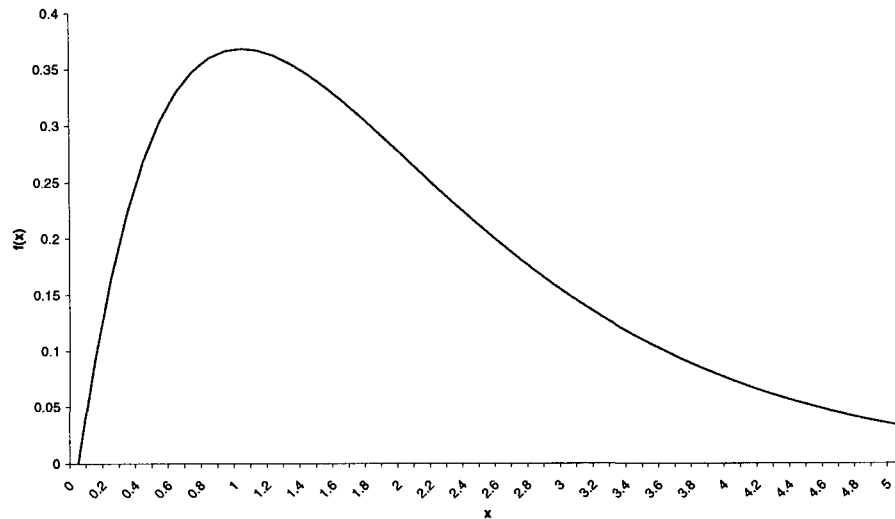


Figura 4.8 Distribución Gamma($\alpha=2, \beta=1$)

- *Generación de variables aleatorias continuas con distribución Weibull*
(α, β)

El pseudocódigo de este algoritmo se obtuvo mediante el método de la transformación y se muestra a continuación:

$$1 - e^{-\frac{x^\gamma}{\beta}} = u$$

$$x = \frac{-\beta \ln(u)}{\gamma}$$

- Weibull (α, β)

$$f(x, \alpha, \beta) = \frac{\alpha}{\beta^\alpha} x^{\alpha-1} e^{-\left(\frac{x}{\beta}\right)^\alpha}, \quad 0 \leq x < \infty, \quad \beta > 0, \quad \alpha > 0$$

$$E(x) = \frac{\beta}{\alpha} \Gamma\left(\frac{1}{\alpha}\right) \quad V(x) = \frac{\beta^2}{\alpha} \left[2\Gamma\left(\frac{2}{\alpha}\right) - \frac{1}{\alpha} \left[\Gamma\left(\frac{1}{\alpha}\right) \right]^2 \right]$$

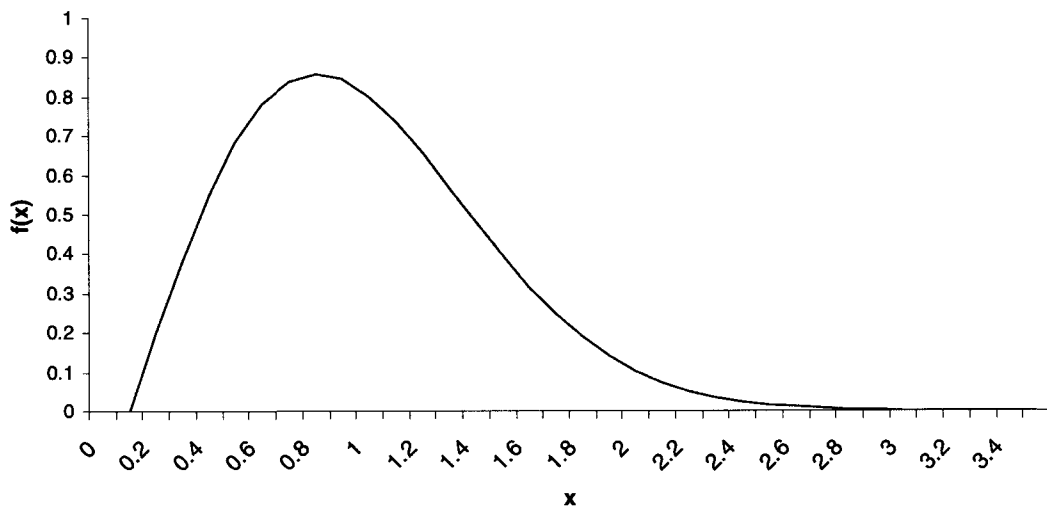


Figura 4.9 Distribución Weibull($\alpha=2, \beta=1$)

4.3 Desplazamiento y escalamiento de las muestras aleatorias

Todas las distribuciones fueron “desplazadas y escaladas” durante las simulaciones para que cada una de ellas tuviera un valor esperado de μ_0 y una varianza de σ_0^2

Se aplicó una transformación lineal a las variables aleatorias continuas utilizando el modelo lineal genérico $X=aY+b$.

De esta forma, si Y es una variable aleatoria continua que proviene de cierta distribución con $E(Y) = \mu'$ y $V(Y) = \sigma'^2$, entonces:

$$E(X) = E(aY + b) = E(aY) + E(b) = aE(Y) + b = a\mu' + b$$

$$V(X) = V(aY + b) = V(aY) = a^2V(Y) = a^2\sigma'^2$$

Se igualan las medias y las varianzas:

$$E(X) = \mu_0 = a\mu' + b$$

$$V(X) = \sigma_0^2 = a^2\sigma'^2$$

Se despejan:

$$a = \frac{\sigma_0}{\sigma'}$$

$$b = \mu_0 - \frac{\sigma_0}{\sigma'}\mu'$$

La transformación lineal queda:

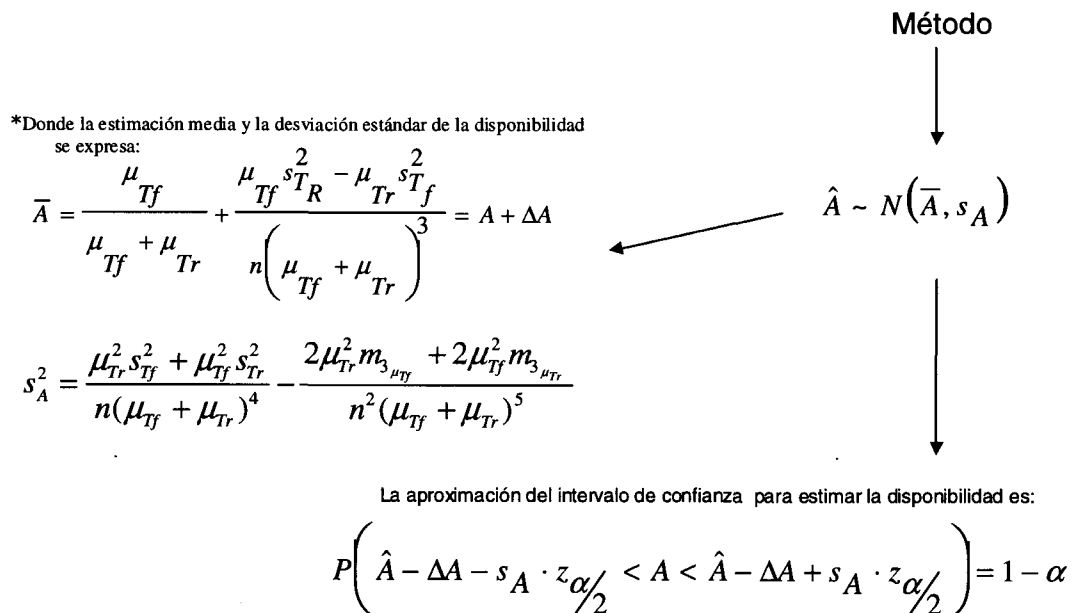
$$X = aY + b = \frac{\sigma_0}{\sigma'}Y + \mu_0 - \frac{\sigma_0}{\sigma'}\mu' = \frac{\sigma_0}{\sigma'}(Y - \mu') + \mu_0$$

Para obtener el valor de la variable aleatoria continua X , que proviene de cierta distribución, con media μ_0 y varianza σ_0^2 , sólo es necesario sustituir el valor de la variable aleatoria continua original Y , así como su media μ' y su varianza σ'^2 , en la ecuación anterior.

4.4 Equivalencia entre simulaciones

Para determinar si la simulación Bootstrap es consistente con el método propuesto por Wang y Kececioglu (2000), se compararon los resultados arrojados por ambos métodos.

Enseguida, se muestra el algoritmo del método propuesto por Wang y Kececioglu (2000) utilizado para obtener los intervalos de confianza de la disponibilidad.



Para realizar las simulaciones antes mencionadas se utilizó el lenguaje de programación C [Gottfried (1997)]. Los algoritmos empleados aparecen en el Anexo D.

Para determinar si ambas simulaciones desarrolladas son consistentes, se compararon los resultados arrojados por ambas simulaciones, los cuales fueron condensados en forma de tabla. Los datos comparados son los que aparecen en las tablas de porcentajes de cobertura para intervalos de confianza del 90% y del 95% que se muestran en el Anexo B.

Como resultado final de la simulación desarrollada en este trabajo de tesis, para los dos métodos *Bootstrap* y *Wang y Kececioglu (2000)*, se obtuvieron 36 tablas (combinaciones de las 6 distribuciones) con dos intervalos de confianza, dando 72 tablas de las cuales se arrojaron 8 valores por tabla (para los casos con $n=20$, $n=30$, $n=50$, $n=100$). En la siguiente figura se observa el número de tablas resultantes de ambas simulaciones:

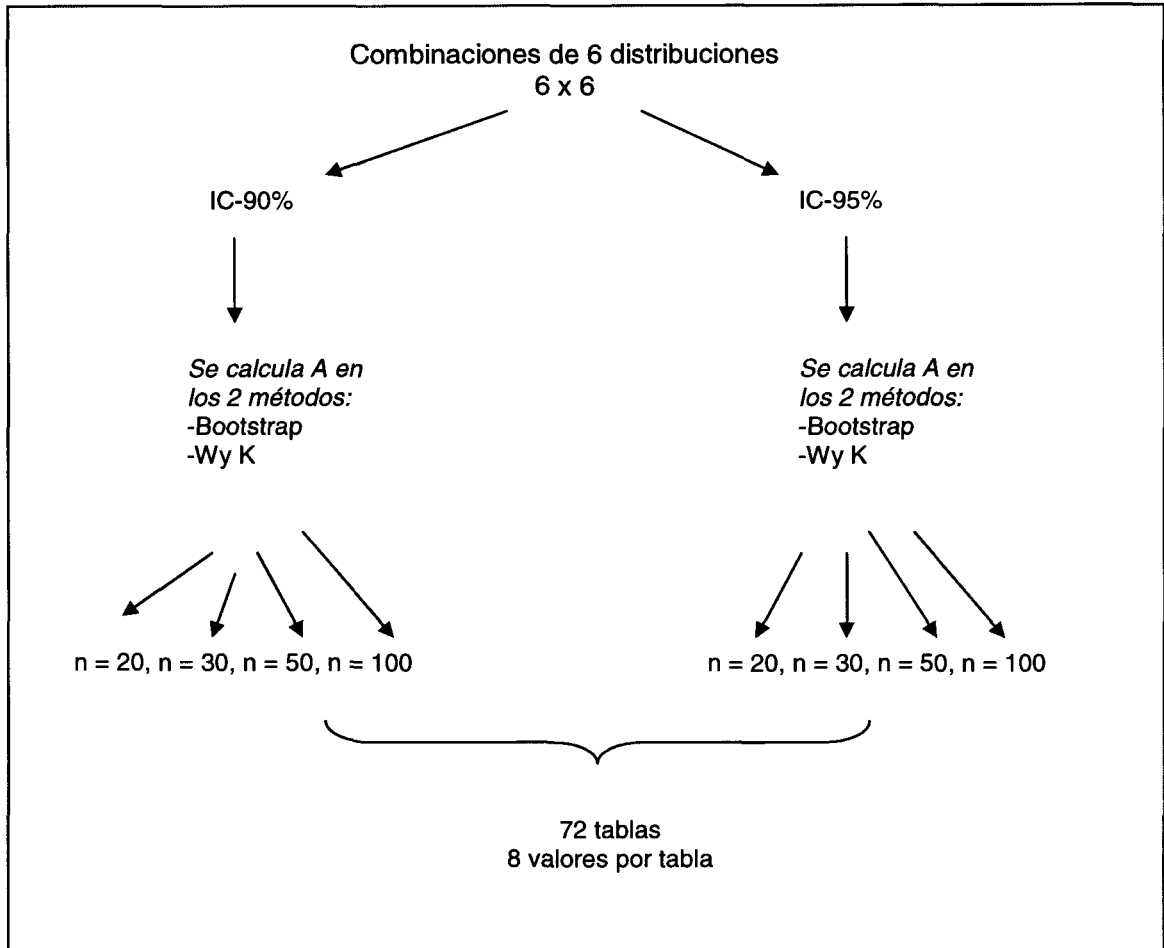


Figura 4.10 Número de tablas resultantes de la simulación.

4.6 Validación del código de simulación: Ejemplo de la distribución Exponencial

Para probar la consistencia de resultados entre la simulación de Wang y Kececioglu (2000) y la simulación desarrollada en este trabajo de tesis, se realizó una prueba de igualdad de proporciones. La idea general fue la comparación de los resultados arrojados en ambos métodos.

Esta prueba para validar igualdad de proporciones se plantea las siguientes hipótesis:

$$H_0 = p_x = p_y$$

$$H_a = p_x \neq p_y$$

Se definen las estimaciones de las proporciones mencionadas en las hipótesis anteriores de la siguiente manera:

$$\hat{p}_x = \frac{x}{n_x}$$

$$\hat{p}_y = \frac{y}{n_y}$$

En las ecuaciones anteriores, x representa el número de observaciones-Intervalos de confianza que incluyen el valor de A , mediante la simulación Wang y Kececioglu y n_x es el total de iteraciones de dicha simulación. Entonces y representa el número de intervalos de confianza que incluyen el valor de A , mediante la simulación Bootstrap, y n_y es el total de iteraciones de dicha simulación.

Esta prueba de hipótesis se utilizó con un nivel de significancia de $\alpha= 0.10$ y $\alpha=0.05$, según sea el caso en la simulación.

En esta prueba para la igualdad de dos proporciones, nos muestra en el Anexo A, que no existe evidencia estadística para rechazar una sola

hipótesis nula, ya que ninguno de los resultados de las pruebas aplicadas a las tablas mostradas en el Anexo A cae dentro de la región de rechazo, lo que nos indica que cada par de proporciones en cada tabla es equivalente para ambas simulaciones.

4.5 Prueba para medir el porcentaje de cobertura

El interés por comparar los resultados arrojados en ambas simulaciones nos permite emplear una prueba de hipótesis que compare en forma individual el porcentaje de cobertura contra un nivel de confianza dado. La comparación del método se basa en la siguiente prueba de hipótesis:

H₀: nivel de confianza = 1 - α

H_a: nivel de confianza < 1 - α

donde:

p es el estadístico y representa el porcentaje de cobertura dado en cada simulación.

Para realizar esta prueba de hipótesis se utilizará la distribución de probabilidad binomial. Con un nivel de significancia de $\alpha = 0.10$ y $\alpha = 0.05$, según sea el caso en la simulación. Se obtendrá el p-valor de cada uno de los porcentajes de cobertura y se realizará la comparación con el valor α .

En el Anexo C, se encuentran los p-valor y se identifica por medio de un asterisco aquellos porcentajes de cobertura que son significativos. Como podemos observar en las tablas del Anexo C, la prueba que obtuvo mayor pruebas significativas en el porcentaje de cobertura es el método Wang y Kececioglu (2000), lo que nos indica que, existe mayor número de porcentajes de cobertura que están por debajo del nivel de confianza dado.

Capítulo 5

Conclusiones

En esta tesis se trabajó con dos métodos para estimar la disponibilidad inherente de un sistema reparable. El método Wang y Kececioglu fue propuesto en el año 2000 [Wang y Kececioglu (2000)], dando un intervalo de confianza mediante la aproximación normal. En este trabajo de tesis se estimó la disponibilidad por medio del intervalo de confianza utilizando la simulación de muestreo Bootstrap.

Para realizar una comparación entre ambas metodologías se llevó a cabo una serie de simulaciones, desarrolladas en el lenguaje de programación C [Gottfried (1997)], las cuales nos proporcionaron los porcentajes de cobertura para cada una de las combinaciones realizadas a partir de las 6 distribuciones utilizadas. El porcentaje de cobertura es definido como el porcentaje de los intervalos de confianza que incluyen al valor real de la disponibilidad. Para validar la simulación, se realizó una comparación por medio de la prueba de dos proporciones, comparando los porcentajes de cobertura del método Wang y Kececioglu (2000) y el método no paramétrico Bootstrap, esta prueba se realizó únicamente, cuando los tiempos entre fallas y los tiempos de reparación se distribuyen exponencialmente. Dicha

prueba de hipótesis se realizó en el capítulo 4, sección 4.5, como se observan los valores nos indican que la cobertura de los dos métodos es equivalente para ambas simulaciones.

Posteriormente, se realizó una prueba de hipótesis para medir el porcentaje de cobertura de forma individual, comparándolo con el valor de significancia dado. Para realizar esta prueba de hipótesis se realizó la prueba binomial, en donde x es el número de intervalos que incluyen a la disponibilidad A mediante el método Wang y Kececioglu (2000), y es el número de intervalos que incluyen a la disponibilidad A mediante el método Bootstrap. Los resultados de esta prueba se encuentran en el Anexo C. En dicha prueba se realizó una comparación del P-valor con el valor de α , lo que nos ayudó a tomar la decisión de rechazar o no la hipótesis nula, para cada una de los porcentajes de cobertura.

En el Anexo C se indica con un (*) si la prueba es o no significativa, esta prueba se realizó con un nivel de significancia de $\alpha=0.05$. Como podemos observar en las tablas del Anexo C, existe un mayor número de porcentajes de cobertura que están por abajo del nivel de confianza dado, para el método Wang y Kececioglu (2000).

También podemos observar que conforme aumenta el tamaño de la muestra para las simulaciones el porcentaje de cobertura se ajusta con el nivel de confianza propuesto.

Como podemos observar en las tablas del anexo C, cuando los tiempos se distribuyen log normal, existe evidencia suficiente para decir que el valor de cobertura está por abajo del deseado.

A través de los resultados obtenidos en las pruebas realizadas en ambas simulaciones, concluimos que la técnica Bootstrap estándar tiene un mejor desempeño que la técnica desarrollada por Wang y Kececioglu. Ya que se ajusta el porcentaje de cobertura con el nivel de confianza propuesto, notándose claramente en las tablas que el desempeño del método Bootstrap mejora notablemente conforme aumenta el tamaño de la muestra. (Anexo C)

El método Bootstrap no pretende ser un sustituto de las técnicas estadísticas tradicionales. Este nuevo tipo de técnicas están pensadas para situaciones en donde las pruebas inferenciales son difíciles de realizar.

Una buena parte del crédito que se dé a las técnicas Bootstrap dependerá de la confianza otorgada al supuesto fundamental en que se basan y que Lunneborg (1983) menciona de la siguiente manera:

“La presunción del Bootstrap es que la distribución muestral del estadístico que se está estudiando y la distribución muestral encontrada mediante este proceso iterativo son, para una amplia gama de estadísticos, esencialmente idénticas”

5.1 Trabajo a futuro

El trabajo de investigación realizado puede extenderse al empleo de otras distribuciones y de esta forma determinar el desempeño del método Bootstrap estándar (técnica no paramétrica) en comparación con la metodología empleada por Wang y Kececioglu (2000). También se puede trabajar con el método Bootstrap percentil [Efron y Tibshirani (1993)], para determinar el desempeño en comparación con el método Bootstrap estándar.

Se puede desarrollar otra técnica no paramétrica para calcular la disponibilidad, cuando los tiempos son distribuidos lognormal.

El algoritmo empleado puede servir como base para nuevas investigaciones que tengan relación con el método Bootstrap debido a su diseño detallado. Estos algoritmos pueden ser utilizados prácticamente en cualquier arquitectura computacional debido a que fueron realizados empleando el lenguaje de programación ANSI C.

Anexo A

Validación del código de simulación: Ejemplo de la distribución Exponencial

Esta prueba para validar igualdad de proporciones se plantea las siguientes hipótesis:

$$H_0 = p_x = p_y$$

$$H_a = p_x \neq p_y$$

Las estimaciones de las proporciones mencionadas en las hipótesis anteriores se definen como:

$$\hat{p}_x = \frac{x}{n_x}$$

$$\hat{p}_y = \frac{y}{n_y}$$

Por el teorema central del límite tenemos:

$$z \equiv \frac{(\hat{p}_x - \hat{p}_y) - 0}{\sqrt{\frac{\bar{p}(1-\bar{p})}{n_x} + \frac{\bar{p}(1-\bar{p})}{n_y}}}$$

donde:

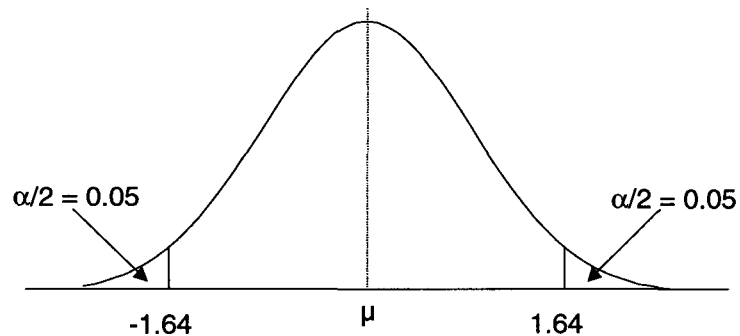
$$\bar{p} = \frac{(n_x \cdot \hat{p}_x) + (n_y \cdot \hat{p}_y)}{n_x + n_y}$$

En la variable \bar{p} , representa una proporción ponderada, para este caso particular de tesis en donde $n_x = n_y = N$, se tiene que:

$$\bar{p} = \frac{\hat{p}_x + \hat{p}_y}{2}$$

Las regiones de rechazo o aceptación para cierto valor de α , si $|Z| > Z_{\alpha/2}$, se rechaza H_0 . De lo contrario no se rechaza.

La región de rechazo para el valor empleado de $\alpha = 0.1$, se muestra en la siguiente figura:



Región de rechazo bilateral para una distribución normal estándar con $\alpha = 0.1$

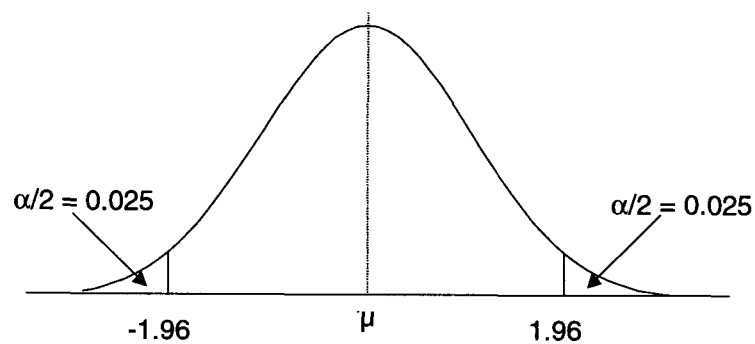
Como se observa en la figura anterior la región de rechazo de la prueba abarca valores menores de -1.64 y valores mayores 1.64 . En la siguiente tabla se muestran los porcentajes de cobertura obtenidos en ambas simulaciones, para un nivel de confianza del 90%.

Distribución	Simulación	Tamaño de muestra			
		n = 20	n = 30	n = 50	n = 100
Tiempo entre falla~Exp Tiempo en reparación~Exp	W y K	0.850	0.864	0.879	0.864
	Bootstrap	0.891	0.891	0.895	0.880
Valores Z		-0.1506	-0.12034	-0.0909	-0.1311
P-valor		0.8802	0.9042	0.9275	0.8956

Como observamos en la tabla, no existe evidencia estadística para rechazar una hipótesis nula, ya que ningún valor cae dentro de la región de rechazo.

Lo que indica que cada par de proporciones en cada tabla es equivalente para ambas simulaciones.

La región de rechazo para el valor empleado de $\alpha = 0.05$, se muestra en la siguiente figura:



Región de rechazo bilateral para una distribución normal estándar con $\alpha = 0.05$

Como se observa en la figura anterior la región de rechazo de la prueba abarca valores menores de -1.96 y valores mayores 1.96. Esta prueba de igualdad de proporciones tiene un nivel de confianza del 95%.

Distribución	Simulación	Tamaño de muestra			
		n = 20	n = 30	n = 50	n = 100
Tiempo entre falla~Exp Tiempo en reparación~Exp	W y K	0.887	0.925	0.925	0.926
	Bootstrap	0.913	0.946	0.936	0.933
Valores Z		-0.0919	-0.0871	-0.05927	-0.0534
P-valor		0.9267	0.9305	0.9527	0.9574

Comparando los valores obtenidos en la tabla anterior, podemos observar que no se rechaza H_0 , para ninguno de los valores, por lo tanto no existe diferencias significativas entre el porcentaje de cobertura del método Wang y Kececioglu (2000) y el método no paramétrico Bootstrap estándar.

Anexo B

Tablas de porcentajes de Cobertura

A continuación se muestran los porcentajes de cobertura para las simulaciones realizadas de Wang y Kececioglu (2000) y Bootstrap.

En las tablas: $Tf \sim$; significa los tiempos entre fallas se distribuye y $Tr \sim$; significa el tiempo de reparación se distribuye.

Distribución Normal

Porcentajes de cobertura para Intervalos de confianza del 90%

Distribución	Simulación	Tamaño de muestra n			
		20	30	50	100
$Tf \sim \text{Normal} - Tr \sim \text{Normal}$	W y K	0.900	0.900	0.902	0.916
	Bootstrap	0.927	0.922	0.913	0.924
$Tf \sim \text{Normal} - Tr \sim \text{Exponencial}$	W y K	0.856	0.886	0.879	0.888
	Bootstrap	0.889	0.917	0.904	0.898
$Tf \sim \text{Normal} - Tr \sim T \text{ Student}$	W y K	0.904	0.909	0.915	0.903
	Bootstrap	0.926	0.930	0.923	0.904
$Tf \sim \text{Normal} - Tr \sim \text{Lognormal}$	W y K	0.830	0.857	0.875	0.894
	Bootstrap	0.877	0.891	0.894	0.899
$Tf \sim \text{Normal} - Tr \sim \text{Gamma}$	W y K	0.886	0.894	0.883	0.906
	Bootstrap	0.910	0.910	0.906	0.913
$Tf \sim \text{Normal} - Tr \sim \text{Weibull}$	W y K	0.900	0.906	0.891	0.888
	Bootstrap	0.922	0.923	0.902	0.895

Porcentajes de cobertura para Intervalos de confianza del 95%

Distribución	Simulación	Tamaño de muestra			
		20	30	50	100
$Tf \sim \text{Normal} - Tr \sim \text{Normal}$	W y K	0.938	0.955	0.949	0.957
	Bootstrap	0.959	0.968	0.954	0.958
$Tf \sim \text{Normal} - Tr \sim \text{Exponencial}$	W y K	0.900	0.934	0.940	0.944
	Bootstrap	0.918	0.945	0.951	0.945
$Tf \sim \text{Normal} - Tr \sim T \text{ Student}$	W y K	0.939	0.945	0.946	0.945
	Bootstrap	0.954	0.958	0.957	0.949
$Tf \sim \text{Normal} - Tr \sim \text{Lognormal}$	W y K	0.879	0.894	0.912	0.925
	Bootstrap	0.911	0.908	0.921	0.931
$Tf \sim \text{Normal} - Tr \sim \text{Gamma}$	W y K	0.927	0.925	0.942	0.949
	Bootstrap	0.944	0.938	0.952	0.953
$Tf \sim \text{Normal} - Tr \sim \text{Weibull}$	W y K	0.929	0.936	0.939	0.947
	Bootstrap	0.948	0.953	0.954	0.952

Distribución Exponencial

Porcentajes de cobertura para Intervalos de confianza del 90%

Distribución	Simulación	Tamaño de muestra			
		20	30	50	100
$Tf \sim \text{Exponencial} - Tr \sim \text{Normal}$	W y K	0.885	0.870	0.876	0.883
	Bootstrap	0.911	0.893	0.894	0.893
$Tf \sim \text{Exponen.} - Tr \sim \text{Exponen.}$	W y K	0.850	0.864	0.879	0.864
	Bootstrap	0.891	0.891	0.895	0.880
$Tf \sim \text{Exponencial} - Tr \sim T \text{ Student}$	W y K	0.861	0.878	0.880	0.898
	Bootstrap	0.892	0.895	0.899	0.906
$Tf \sim \text{Exponencial} - Tr \sim \text{Lognormal}$	W y K	0.804	0.842	0.846	0.892
	Bootstrap	0.852	0.867	0.866	0.900
$Tf \sim \text{Exponencial} - Tr \sim \text{Gamma}$	W y K	0.859	0.869	0.885	0.885
	Bootstrap	0.888	0.883	0.899	0.892
$Tf \sim \text{Exponencial} - Tr \sim \text{Weibull}$	W y K	0.839	0.873	0.874	0.898
	Bootstrap	0.876	0.894	0.889	0.898

Porcentajes de cobertura para Intervalos de confianza del 95%

Distribución	Simulación	Tamaño de muestra			
		20	30	50	100
$Tf \sim \text{Exponencial} - Tr \sim \text{Normal}$	W y K	0.918	0.938	0.937	0.948
	Bootstrap	0.939	0.949	0.946	0.948
$Tf \sim \text{Exponen.} - Tr \sim \text{Exponen.}$	W y K	0.887	0.925	0.925	0.926
	Bootstrap	0.913	0.946	0.936	0.933
$Tf \sim \text{Exponencial} - Tr \sim T \text{ student}$	W y K	0.937	0.926	0.940	0.945
	Bootstrap	0.959	0.938	0.948	0.943
$Tf \sim \text{Exponen.} - Tr \sim \text{Lognormal}$	W y K	0.879	0.890	0.926	0.931
	Bootstrap	0.922	0.912	0.942	0.942
$Tf \sim \text{Exponencial} - Tr \sim \text{Gamma}$	W y K	0.906	0.914	0.924	0.940
	Bootstrap	0.933	0.933	0.936	0.949
$Tf \sim \text{Exponencial} - Tr \sim \text{Weibull}$	W y K	0.903	0.934	0.934	0.935
	Bootstrap	0.927	0.953	0.943	0.941

Distribución T de Student

Porcentajes de cobertura para Intervalos de confianza del 90%

Distribución	Simulación	Tamaño de muestra			
		20	30	50	100
<i>Tf</i> ~T Student- <i>Tr</i> ~Normal	W y K	0.887	0.892	0.913	0.911
	Bootstrap	0.914	0.913	0.927	0.915
<i>Tf</i> ~T Student- <i>Tr</i> ~Exponen.	W y K	0.843	0.883	0.881	0.907
	Bootstrap	0.877	0.912	0.898	0.914
<i>Tf</i> ~T Student- <i>Tr</i> ~T Student	W y K	0.898	0.892	0.898	0.903
	Bootstrap	0.914	0.918	0.911	0.906
<i>Tf</i> ~T Student- <i>Tr</i> ~Lognormal	W y K	0.846	0.845	0.866	0.886
	Bootstrap	0.872	0.867	0.880	0.896
<i>Tf</i> ~T Student- <i>Tr</i> ~Gamma	W y K	0.873	0.880	0.908	0.892
	Bootstrap	0.896	0.899	0.924	0.897
<i>Tf</i> ~T Student- <i>Tr</i> ~Weibull	W y K	0.880	0.891	0.913	0.905
	Bootstrap	0.903	0.913	0.926	0.914

Porcentajes de cobertura para Intervalos de confianza del 95%

Distribución	Simulación	Tamaño de muestra			
		20	30	50	100
<i>Tf</i> ~T Student- <i>Tr</i> ~Normal	W y K	0.947	0.940	0.946	0.955
	Bootstrap	0.954	0.952	0.955	0.954
<i>Tf</i> ~T Student- <i>Tr</i> ~Exponen.	W y K	0.909	0.933	0.942	0.943
	Bootstrap	0.937	0.948	0.952	0.947
<i>Tf</i> ~T Student- <i>Tr</i> ~T Student	W y K	0.924	0.939	0.953	0.964
	Bootstrap	0.938	0.959	0.961	0.968
<i>Tf</i> ~T Student- <i>Tr</i> ~Lognormal	W y K	0.873	0.887	0.905	0.934
	Bootstrap	0.909	0.914	0.916	0.940
<i>Tf</i> ~T Student- <i>Tr</i> ~Gamma	W y K	0.912	0.921	0.949	0.958
	Bootstrap	0.932	0.936	0.960	0.963
<i>Tf</i> ~T Student- <i>Tr</i> ~Weibull	W y K	0.927	0.931	0.944	0.951
	Bootstrap	0.947	0.949	0.951	0.955

Distribución Lognormal

Porcentajes de cobertura para Intervalos de confianza del 90%

Distribución	Simulación	Tamaño de muestra			
		20	30	50	100
$Tf \sim \text{Lognormal} - Tr \sim \text{Normal}$	W y K	0.851	0.856	0.873	0.867
	Bootstrap	0.882	0.871	0.891	0.876
$Tf \sim \text{Lognormal} - Tr \sim \text{Exponen.}$	W y K	0.84	0.848	0.855	0.881
	Bootstrap	0.878	0.874	0.873	0.891
$Tf \sim \text{Lognormal} - Tr \sim T \text{ Student}$	W y K	0.838	0.871	0.872	0.897
	Bootstrap	0.879	0.891	0.892	0.896
$Tf \sim \text{Lognormal} - Tr \sim \text{Lognormal}$	W y K	0.795	0.818	0.862	0.86
	Bootstrap	0.845	0.863	0.885	0.879
$Tf \sim \text{Lognormal} - Tr \sim \text{Gamma}$	W y K	0.826	0.865	0.866	0.86
	Bootstrap	0.865	0.887	0.878	0.879
$Tf \sim \text{Lognormal} - Tr \sim \text{Weibull}$	W y K	0.853	0.867	0.872	0.888
	Bootstrap	0.886	0.89	0.886	0.892

Porcentajes de cobertura para Intervalos de confianza del 95%

Distribución	Simulación	Tamaño de muestra			
		20	30	50	100
$Tf \sim \text{Lognormal} - Tr \sim \text{Normal}$	W y K	0.903	0.92	0.942	0.948
	Bootstrap	0.938	0.939	0.952	0.949
$Tf \sim \text{Lognormal} - Tr \sim \text{Exponen.}$	W y K	0.874	0.903	0.919	0.931
	Bootstrap	0.908	0.923	0.933	0.939
$Tf \sim \text{Lognormal} - Tr \sim T \text{ Student}$	W y K	0.886	0.904	0.932	0.935
	Bootstrap	0.921	0.93	0.938	0.944
$Tf \sim \text{Lognormal} - Tr \sim \text{Lognormal}$	W y K	0.871	0.878	0.9	0.944
	Bootstrap	0.914	0.915	0.919	0.947
$Tf \sim \text{Lognormal} - Tr \sim \text{Gamma}$	W y K	0.891	0.91	0.923	0.934
	Bootstrap	0.925	0.931	0.935	0.945
$Tf \sim \text{Lognormal} - Tr \sim \text{Weibull}$	W y K	0.906	0.94	0.928	0.934
	Bootstrap	0.934	0.953	0.94	0.943

Distribución Gamma

Porcentajes de cobertura para Intervalos de confianza del 90%

Distribución	Simulación	Tamaño de muestra			
		20	30	50	100
<i>Tf</i> ~Gamma- <i>Tr</i> ~Normal	W y K	0.897	0.891	0.887	0.899
	Bootstrap	0.917	0.910	0.903	0.904
<i>Tf</i> ~Gamma- <i>Tr</i> ~Exponencial	W y K	0.855	0.879	0.874	0.897
	Bootstrap	0.890	0.904	0.887	0.905
<i>Tf</i> ~Gamma- <i>Tr</i> ~T Student	W y K	0.884	0.872	0.899	0.886
	Bootstrap	0.910	0.896	0.910	0.892
<i>Tf</i> ~Gamma- <i>Tr</i> ~Lognormal	W y K	0.828	0.862	0.877	0.867
	Bootstrap	0.877	0.891	0.894	0.879
<i>Tf</i> ~Gamma- <i>Tr</i> ~Gamma	W y K	0.864	0.896	0.884	0.894
	Bootstrap	0.903	0.921	0.902	0.906
<i>Tf</i> ~Gamma- <i>Tr</i> ~Weibull	W y K	0.867	0.889	0.886	0.917
	Bootstrap	0.898	0.908	0.900	0.923

Porcentajes de cobertura para Intervalos de confianza del 95%

Distribución	Simulación	Tamaño de muestra			
		20	30	50	100
<i>Tf</i> ~Gamma- <i>Tr</i> ~Normal	W y K	0.925	0.927	0.938	0.947
	Bootstrap	0.941	0.940	0.951	0.951
<i>Tf</i> ~Gamma- <i>Tr</i> ~Exponencial	W y K	0.896	0.918	0.934	0.939
	Bootstrap	0.927	0.938	0.943	0.943
<i>Tf</i> ~Gamma- <i>Tr</i> ~T Student	W y K	0.938	0.937	0.943	0.951
	Bootstrap	0.962	0.951	0.953	0.954
<i>Tf</i> ~Gamma- <i>Tr</i> ~Lognormal	W y K	0.872	0.895	0.910	0.924
	Bootstrap	0.913	0.910	0.924	0.937
<i>Tf</i> ~Gamma- <i>Tr</i> ~Gamma	W y K	0.905	0.944	0.935	0.945
	Bootstrap	0.935	0.956	0.946	0.949
<i>Tf</i> ~Gamma- <i>Tr</i> ~Weibull	W y K	0.920	0.918	0.937	0.948
	Bootstrap	0.947	0.939	0.948	0.951

Distribución Weibull

Porcentajes de cobertura para Intervalos de confianza del 90%

Distribución	Simulación	Tamaño de muestra			
		20	30	50	100
Tf ~Weibull- Tr ~Normal	W y K	0.889	0.898	0.884	0.904
	Bootstrap	0.918	0.924	0.905	0.908
Tf ~Weibull- Tr ~Exponencial	W y K	0.869	0.882	0.886	0.897
	Bootstrap	0.906	0.896	0.899	0.904
Tf ~Weibull- Tr ~T Student	W y K	0.875	0.886	0.894	0.892
	Bootstrap	0.909	0.912	0.906	0.892
Tf ~Weibull- Tr ~Lognormal	W y K	0.827	0.852	0.879	0.885
	Bootstrap	0.890	0.892	0.906	0.893
Tf ~Weibull- Tr ~Gamma	W y K	0.867	0.865	0.897	0.903
	Bootstrap	0.895	0.893	0.904	0.913
Tf ~Weibull- Tr ~Weibull	W y K	0.895	0.869	0.887	0.896
	Bootstrap	0.922	0.880	0.898	0.903

Porcentajes de cobertura para Intervalos de confianza del 95%

Distribución	Simulación	Tamaño de muestra			
		20	30	50	100
Tf ~Weibull- Tr ~Normal	W y K	0.947	0.947	0.940	0.938
	Bootstrap	0.962	0.961	0.952	0.945
Tf ~Weibull- Tr ~Exponencial	W y K	0.891	0.917	0.938	0.933
	Bootstrap	0.914	0.930	0.948	0.938
Tf ~Weibull- Tr ~T Student	W y K	0.949	0.941	0.953	0.942
	Bootstrap	0.956	0.950	0.963	0.947
Tf ~Weibull- Tr ~Lognormal	W y K	0.884	0.890	0.909	0.931
	Bootstrap	0.907	0.915	0.928	0.936
Tf ~Weibull- Tr ~Gamma	W y K	0.922	0.934	0.928	0.943
	Bootstrap	0.944	0.947	0.944	0.948
Tf ~Weibull- Tr ~Weibull	W y K	0.918	0.938	0.939	0.938
	Bootstrap	0.941	0.950	0.947	0.940

Anexo C

Tablas del desempeño de la cobertura

*, significa, rechazar la Hipótesis nula. Prueba que se observa en el capítulo 4, sección 4.6. Con un valor de $\alpha = 0.05$

Porcentajes de cobertura para Intervalos de confianza del 90%

Tamaño de muestra $n = 20$

	W y K	p-valor		BS	p-valor
Tf-Normal-Tr-Normal	0.900	0.51541771		0.927	0.9987298
Tf-Normal-Tr-Exponencial	0.856	6.8729E-06	*	0.889	0.1347764
Tf-Normal-Tr-T Student	0.904	0.67844592		0.926	0.9981510
Tf-Normal-Tr-Lognormal	0.830	7.3909E-12	*	0.877	0.0103621
Tf-Normal-Tr-Gamma	0.886	0.07929876		0.910	0.8665962
Tf-Normal-Tr-Weibull	0.900	0.51541771		0.922	0.9927527
Tf-Exponencial-Tr-Normal	0.885	0.06537196		0.911	0.8886772
Tf-Exponencial-Tr-Exponencial	0.850	4.4894E-07	*	0.891	0.1843475
Tf-Exponencial-Tr-T Student	0.861	5.4013E-05	*	0.892	0.2130407
Tf-Exponencial-Tr-Lognormal	0.804	7.3357E-20	*	0.852	1.1492E-06
Tf-Exponencial-Tr-Gamma	0.859	2.424E-05	*	0.888	0.11385744
Tf-Exponencial-Tr-Weibull	0.839	1.5003E-09	*	0.876	0.00793071
Tf-T Student-Tr-Normal	0.887	0.09540801		0.914	0.93930627
Tf-T Student-Tr-Exponencial	0.843	1.3218E-08	*	0.877	0.01036217
Tf-T Student-Tr-T Student	0.898	0.43180014		0.914	0.93930627
Tf-T Student-Tr-Lognormal	0.846	6.261E-08	*	0.872	0.00250628
Tf-T Student-Tr-Gamma	0.873	0.00338412	*	0.896	0.3515951
Tf-T Student-Tr-Weibull	0.880	0.02199614	*	0.903	0.63934037

Porcentajes de cobertura para Intervalos de confianza del 90%

Tamaño de muestra n = 20

	W y K	p-valor		BS	p-valor	
Tf-lognormal-Tr-Normal	0.851	7.2101E-07	*	0.882	0.03485968	*
Tf-lognormal-Tr-Exponencial	0.840	2.6131E-09	*	0.878	0.0134278	*
Tf-lognormal-Tr-T Student	0.838	8.5528E-10	*	0.879	0.01725723	*
Tf-lognormal-Tr-Lognormal	0.795	4.5479E-23	*	0.845	3.7556E-08	*
Tf-lognormal-Tr-Gamma	0.826	5.814E-13	*	0.865	0.00024395	*
Tf-lognormal-Tr-Weibull	0.853	1.8179E-06	*	0.886	0.07929876	
Tf-Gamma -Tr-Normal	0.897	0.39106037		0.917	0.97007297	
Tf-Gamma-Tr-Exponencial	0.855	4.4458E-06	*	0.890	0.15825753	
Tf-Gamma -Tr-T Student	0.884	0.05345008		0.910	0.86659623	
Tf-Gamma-Tr-Lognormal	0.828	2.1016E-12	*	0.877	0.01036217	*
Tf-Gamma-Tr-Gamma	0.864	0.00016935	*	0.903	0.63934037	
Tf-Gamma-Tr-Weibull	0.867	0.0004942	*	0.898	0.43180014	
Tf-Weibull-Tr-Normal	0.889	0.13477649		0.918	0.97690333	
Tf-Weibull-Tr-Exponencial	0.869	0.00096944	*	0.906	0.75101218	
Tf-Weibull -Tr-T Student	0.875	0.00601997	*	0.909	0.84176186	
Tf-Weibull -Tr-Lognormal	0.827	1.1092E-12	*	0.890	0.15825753	
Tf-Weibull-Tr-Gamma	0.867	0.0004942	*	0.895	0.31377421	
Tf-Weibull-Tr-Weibull	0.895	0.31377421		0.922	0.99275277	

Porcentajes de cobertura para Intervalos de confianza del 95%

Tamaño de muestra n = 20

	W y K	p-valor	BS	p-valor
Tf-Normal-Tr-Normal	0.938	0.05110956	0.959	0.91936343
Tf-Normal-Tr-Exponencial	0.900	8.4103E-11 *	0.918	1.188E-05 *
Tf-Normal-Tr-T Student	0.939	0.06706252	0.954	0.73903568
Tf-Normal-Tr-Lognormal	0.879	1.0092E-18 *	0.911	1.8527E-07 *
Tf-Normal-Tr-Gamma	0.927	0.00100598 *	0.944	0.21007608
Tf-Normal-Tr-Weibull	0.929	0.0023303 *	0.948	0.40581607
Tf-Exponencial-Tr-Normal	0.918	1.188E-05 *	0.939	0.06706252
Tf-Exponencial-Tr-Exponencial	0.887	1.7588E-15 *	0.913	6.4634E-07 *
Tf-Exponencial-Tr-T Student	0.937	0.03839324 *	0.959	0.91936343
Tf-Exponencial-Tr-Lognormal	0.879	1.0092E-18 *	0.922	9.7337E-05 *
Tf-Exponencial-Tr-Gamma	0.906	6.6369E-09 *	0.933	0.01059131 *
Tf-Exponencial-Tr-Weibull	0.903	7.8526E-10 *	0.927	0.00100598 *
Tf-T Student-Tr-Normal	0.947	0.35139756	0.954	0.73903568
Tf-T Student-Tr-Exponencial	0.909	5.065E-08 *	0.937	0.03839324 *
Tf-T Student-Tr-T Student	0.924	0.00025781 *	0.938	0.05110956
Tf-T Student-Tr-Lognormal	0.873	2.5324E-21 *	0.909	5.065E-08 *
Tf-T Student-Tr-Gamma	0.912	3.4811E-07 *	0.932	0.00740772 *
Tf-T Student-Tr-Weibull	0.927	0.00100598 *	0.947	0.35139756

	W y K	p-valor	BS	p-valor
Tf-lognormal-Tr-Normal	0.903	7.8526E-10 *	0.938	0.05110956
Tf-lognormal-Tr-Exponencial	0.874	7.0266E-21 *	0.908	2.6024E-08 *
Tf-lognormal-Tr-T Student	0.886	7.1552E-16 *	0.921	5.865E-05 *
Tf-lognormal-Tr-Lognormal	0.871	3.2031E-22 *	0.914	1.1857E-06 *
Tf-lognormal-Tr-Gamma	0.891	5.8191E-14 *	0.925	0.00041134 *
Tf-lognormal-Tr-Weibull	0.906	6.6369E-09 *	0.934	0.01493041 *
Tf-Gamma-Tr-Normal	0.925	0.00041134 *	0.941	0.11056152
Tf-Gamma-Tr-Exponencial	0.896	3.6789E-12 *	0.927	0.00100598 *
Tf-Gamma-Tr-T Student	0.938	0.05110956	0.962	0.96933561
Tf-Gamma-Tr-Lognormal	0.872	9.0463E-22 *	0.913	6.4634E-07 *
Tf-Gamma-Tr-Gamma	0.905	3.295E-09 *	0.935	0.0207499 *
Tf-Gamma-Tr-Weibull	0.920	3.4886E-05 *	0.947	0.35139756
Tf-Weibull-Tr-Normal	0.947	0.35139756	0.962	0.96933561
Tf-Weibull-Tr-Exponencial	0.891	5.8191E-14 *	0.914	1.1857E-06 *
Tf-Weibull-Tr-T Student	0.949	0.46247096	0.956	0.82670891
Tf-Weibull-Tr-Lognormal	0.884	1.1503E-16 *	0.907	1.3218E-08 *
Tf-Weibull-Tr-Gamma	0.922	9.7337E-05 *	0.944	0.21007608
Tf-Weibull-Tr-Weibull	0.918	1.188E-05 *	0.941	0.11056152

Porcentajes de cobertura para Intervalos de confianza del 90%

Tamaño de muestra n = 30

	W y K	p-valor	BS	p-valor
Tf-Normal-Tr-Normal	0.900	0.5154177	0.922	0.992752772
Tf-Normal-Tr-Exponencial	0.886	0.0792988	0.917	0.970072965
Tf-Normal-Tr-T Student	0.909	0.8417619	0.930	0.999621898
Tf-Normal-Tr-Lognormal	0.857	1.054E-05 *	0.891	0.184347576
Tf-Normal-Tr-Gamma	0.894	0.2779144	0.910	0.866596228
Tf-Normal-Tr-Weibull	0.906	0.7510122	0.923	0.994745606
Tf-Exponencial-Tr-Normal	0.870	0.0013414 *	0.893	0.244272347
Tf-Exponencial-Tr-Exponencial	0.864	0.0001693 *	0.891	0.184347576
Tf-Exponencial-Tr-T Student	0.878	0.0134278 *	0.895	0.31377421
Tf-Exponencial-Tr-Lognormal	0.842	7.756E-09 *	0.867	0.000494198 *
Tf-Exponencial-Tr-Gamma	0.869	0.0009694 *	0.883	0.043343891 *
Tf-Exponencial-Tr-Weibull	0.873	0.0033841 *	0.894	0.277914406
Tf-T Student-Tr-Normal	0.892	0.2130408	0.913	0.924894164
Tf-T Student-Tr-Exponencial	0.883	0.0433439 *	0.912	0.908070838
Tf-T Student-Tr-T Student	0.892	0.2130408	0.918	0.976903326
Tf-T Student-Tr-Lognormal	0.845	3.756E-08 *	0.867	0.000494198 *
Tf-T Student-Tr-Gamma	0.880	0.0219961 *	0.899	0.473400919
Tf-T Student-Tr-Weibull	0.891	0.1843476	0.913	0.924894164

	W y K	p-valor	BS	p-valor
Tf-lognormal-Tr-Normal	0.856	7E-06 *	0.871	0.001841042 *
Tf-lognormal-Tr-Exponencial	0.848	2E-07 *	0.874	0.004532135 *
Tf-lognormal-Tr-T Student	0.871	0.002 *	0.891	0.184347576
Tf-lognormal-Tr-Lognormal	0.818	3E-15 *	0.863	0.000116625 *
Tf-lognormal-Tr-Gamma	0.865	2E-04 *	0.887	0.095408006
Tf-lognormal-Tr-Weibull	0.867	5E-04 *	0.890	0.15825753
Tf-Gamma-Tr-Normal	0.891	0.184	0.910	0.866596228
Tf-Gamma-Tr-Exponencial	0.879	0.017 *	0.904	0.678445923
Tf-Gamma-Tr-T Student	0.872	0.003 *	0.896	0.351595097
Tf-Gamma-Tr-Lognormal	0.862	8E-05 *	0.891	0.184347576
Tf-Gamma-Tr-Gamma	0.896	0.352	0.921	0.990132564
Tf-Gamma-Tr-Weibull	0.889	0.135	0.908	0.814168115
Tf-Weibull-Tr-Normal	0.898	0.432	0.924	0.996240232
Tf-Weibull-Tr-Exponencial	0.882	0.035 *	0.896	0.351595097
Tf-Weibull-Tr-T Student	0.886	0.079	0.912	0.908070838
Tf-Weibull-Tr-Lognormal	0.852	1E-06 *	0.892	0.213040777
Tf-Weibull-Tr-Gamma	0.865	2E-04 *	0.893	0.244272347
Tf-Weibull-Tr-Weibull	0.869	1E-03 *	0.880	0.021996145 *

Porcentajes de cobertura para Intervalos de confianza del 95%

Tamaño de muestra n = 30

	W y K	p-valor		BS	Z
Tf-Normal-Tr-Normal	0.955	0.7853171		0.968	0.997804198
Tf-Normal-Tr-Exponencial	0.934	0.0149304	*	0.945	0.252881914
Tf-Normal-Tr-T Student	0.945	0.2528819		0.958	0.893703829
Tf-Normal-Tr-Lognormal	0.894	7.224E-13	*	0.908	2.60237E-08
Tf-Normal-Tr-Gamma	0.925	0.0004113	*	0.938	0.051109559
Tf-Normal-Tr-Weibull	0.936	0.0284284	*	0.953	0.68846505
Tf-Exponencial-Tr-Normal	0.938	0.0511096		0.949	0.462470959
Tf-Exponencial-Tr-Exponencial	0.925	0.0004113	*	0.946	0.300167429
Tf-Exponencial-Tr-T Student	0.926	0.0006476	*	0.938	0.051109559
Tf-Exponencial-Tr-Lognormal	0.890	2.463E-14	*	0.912	3.48113E-07
Tf-Exponencial-Tr-Gamma	0.914	1.186E-06	*	0.933	0.01059131
Tf-Exponencial-Tr-Weibull	0.934	0.0149304	*	0.953	0.68846505
Tf-T Student-Tr-Normal	0.940	0.0867322		0.952	0.634439985
Tf-T Student-Tr-Exponencial	0.933	0.0105913	*	0.948	0.405816073
Tf-T Student-Tr-T Student	0.939	0.0670625		0.959	0.919363427
Tf-T Student-Tr-Lognormal	0.887	1.759E-15	*	0.914	1.18569E-06
Tf-T Student-Tr-Gamma	0.921	5.865E-05	*	0.936	0.028428397
Tf-T Student-Tr-Weibull	0.931	0.0051087	*	0.949	0.462470959

	W y K	p-valor		BS	p-valor
Tf-lognormal-Tr-Normal	0.920	3E-05	*	0.939	0.067062519
Tf-lognormal-Tr-Exponencial	0.903	8E-10	*	0.923	0.000159455
Tf-lognormal-Tr-T Student	0.904	2E-09	*	0.930	0.003474359
Tf-lognormal-Tr-Lognormal	0.878	4E-19	*	0.915	2.14885E-06
Tf-lognormal-Tr-Gamma	0.910	1E-07	*	0.931	0.005108727
Tf-lognormal-Tr-Weibull	0.940	0.087		0.953	0.68846505
Tf-Gamma-Tr-Normal	0.927	0.001	*	0.940	0.086732178
Tf-Gamma-Tr-Exponencial	0.918	1E-05	*	0.938	0.051109559
Tf-Gamma-Tr-T Student	0.937	0.038	*	0.951	0.577986161
Tf-Gamma-Tr-Lognormal	0.895	2E-12	*	0.910	9.74397E-08
Tf-Gamma-Tr-Gamma	0.944	0.21		0.956	0.826708912
Tf-Gamma-Tr-Weibull	0.918	1E-05	*	0.939	0.067062519
Tf-Weibull-Tr-Normal	0.947	0.351		0.961	0.956651809
Tf-Weibull-Tr-Exponencial	0.917	7E-06	*	0.930	0.003474359
Tf-Weibull-Tr-T Student	0.941	0.111		0.950	0.520258943
Tf-Weibull-Tr-Lognormal	0.890	2E-14	*	0.915	2.14885E-06
Tf-Weibull-Tr-Gamma	0.934	0.015	*	0.947	0.351397564
Tf-Weibull-Tr-Weibull	0.938	0.051		0.950	0.520258943

Porcentajes de cobertura para Intervalos de confianza del 90%

Tamaño de muestra n = 50

	W y K	p-valor		BS	p-valor
Tf-Normal-Tr-Normal	0.902	0.599		0.913	0.924894164
Tf-Normal-Tr-Exponencial	0.879	0.017	*	0.904	0.678445923
Tf-Normal-Tr-T Student	0.915	0.951		0.923	0.994745606
Tf-Normal-Tr-Lognormal	0.875	0.006	*	0.894	0.277914406
Tf-Normal-Tr-Gamma	0.883	0.043	*	0.906	0.751012182
Tf-Normal-Tr-Weibull	0.891	0.184		0.902	0.598846193
Tf-Exponencial-Tr-Normal	0.876	0.008	*	0.894	0.277914406
Tf-Exponencial-Tr-Exponencial	0.879	0.017	*	0.895	0.31377421
Tf-Exponencial-Tr-T Student	0.880	0.022	*	0.899	0.473400919
Tf-Exponencial-Tr-Lognormal	0.846	6E-08	*	0.866	0.00034861 *
Tf-Exponencial-Tr-Gamma	0.885	0.065	*	0.899	0.473400919
Tf-Exponencial-Tr-Weibull	0.874	0.005	*	0.889	0.134776489
Tf-T Student-Tr-Normal	0.913	0.925		0.927	0.998729892
Tf-T Student-Tr-Exponencial	0.881	0.028	*	0.898	0.431800136
Tf-T Student-Tr-T Student	0.898	0.432		0.911	0.888677281
Tf-T Student-Tr-Lognormal	0.866	3E-04	*	0.880	0.021996145 *
Tf-T Student-Tr-Gamma	0.908	0.814		0.924	0.996240232
Tf-T Student-Tr-Weibull	0.913	0.925		0.926	0.998151084

	W y K	p-valor		BS	p-valor
Tf-lognormal-Tr-Normal	0.873	0.003	*	0.891	0.184347576
Tf-lognormal-Tr-Exponencial	0.855	4E-06	*	0.873	0.003384116 *
Tf-lognormal-Tr-T Student	0.872	0.003	*	0.892	0.213040777
Tf-lognormal-Tr-Lognormal	0.862	8E-05	*	0.885	0.065371963
Tf-lognormal-Tr-Gamma	0.866	3E-04	*	0.878	0.013427805 *
Tf-lognormal-Tr-Weibull	0.872	0.003	*	0.886	0.079298764
Tf-Gamma-Tr-Normal	0.887	0.095		0.903	0.639340372
Tf-Gamma-Tr-Exponencial	0.874	0.005	*	0.887	0.095408006
Tf-Gamma-Tr-T Student	0.899	0.473		0.910	0.866596228
Tf-Gamma-Tr-Lognormal	0.877	0.01	*	0.894	0.277914406
Tf-Gamma-Tr-Gamma	0.884	0.053		0.902	0.598846193
Tf-Gamma-Tr-Weibull	0.886	0.079		0.900	0.51541771
Tf-Weibull-Tr-Normal	0.884	0.053		0.905	0.715779842
Tf-Weibull-Tr-Exponencial	0.886	0.079		0.899	0.473400919
Tf-Weibull-Tr-T Student	0.894	0.278		0.906	0.751012182
Tf-Weibull-Tr-Lognormal	0.879	0.017	*	0.906	0.751012182
Tf-Weibull-Tr-Gamma	0.897	0.391		0.904	0.678445923
Tf-Weibull-Tr-Weibull	0.887	0.095		0.898	0.431800136

Porcentajes de cobertura para Intervalos de confianza del 95%

Tamaño de muestra n = 50

	W y K	p-valor	BS	p-valor
Tf~Normal-Tr~Normal	0.949	0.462	0.954	0.739035683
Tf~Normal-Tr~Exponencial	0.940	0.087	0.951	0.577986161
Tf~Normal-Tr~T Student	0.946	0.3	0.957	0.862867295
Tf~Normal-Tr~Lognormal	0.912	3E-07	* 0.921	5.86503E-05 *
Tf~Normal-Tr~Gamma	0.942	0.139	0.952	0.634439985
Tf~Normal-Tr~Weibull	0.939	0.067	0.954	0.739035683
Tf~Exponencial-Tr~Normal	0.937	0.038	* 0.946	0.300167429
Tf~Exponencial-Tr~Exponencial	0.925	4E-04	* 0.936	0.028428397 *
Tf~Exponencial-Tr~T Student	0.940	0.087	0.948	0.405816073
Tf~Exponencial-Tr~Lognormal	0.926	6E-04	* 0.942	0.138918946
Tf~Exponencial-Tr~Gamma	0.924	3E-04	* 0.936	0.028428397 *
Tf~Exponencial-Tr~Weibull	0.934	0.015	* 0.943	0.172057739
Tf~T Student-Tr~Normal	0.946	0.3	0.955	0.785317078
Tf~T Student-Tr~Exponencial	0.942	0.139	0.952	0.634439985
Tf~T Student-Tr~T Student	0.953	0.688	0.961	0.956651809
Tf~T Student-Tr~Lognormal	0.905	3E-09	* 0.916	3.847E-06 *
Tf~T Student-Tr~Gamma	0.949	0.462	0.960	0.940185121
Tf~T Student-Tr~Weibull	0.944	0.21	0.951	0.577986161

	W y K	p-valor	BS	p-valor
Tf~lognormal-Tr~Normal	0.942	0.139	0.952	0.634439985
Tf~lognormal-Tr~Exponencial	0.919	2E-05	* 0.933	0.01059131 *
Tf~lognormal-Tr~T Student	0.932	0.007	* 0.938	0.051109559
Tf~lognormal-Tr~Lognormal	0.900	8E-11	* 0.919	2.04874E-05 *
Tf~lognormal-Tr~Gamma	0.923	2E-04	* 0.935	0.020749899 *
Tf~lognormal-Tr~Weibull	0.928	0.002	* 0.940	0.086732178
Tf~Gamma-Tr~Normal	0.938	0.051	0.951	0.577986161
Tf~Gamma-Tr~Exponencial	0.934	0.015	* 0.943	0.172057739
Tf~Gamma-Tr~T Student	0.943	0.172	0.953	0.68846505
Tf~Gamma-Tr~Lognormal	0.910	1E-07	* 0.924	0.000257807 *
Tf~Gamma-Tr~Gamma	0.935	0.021	* 0.946	0.300167429
Tf~Gamma-Tr~Weibull	0.937	0.038	* 0.948	0.405816073
Tf~Weibull-Tr~Normal	0.940	0.087	0.952	0.634439985
Tf~Weibull-Tr~Exponencial	0.938	0.051	0.948	0.405816073
Tf~Weibull-Tr~T Student	0.953	0.688	0.963	0.978845168
Tf~Weibull-Tr~Lognormal	0.909	5E-08	* 0.928	0.001541587 *
Tf~Weibull-Tr~Gamma	0.928	0.002	* 0.944	0.210076079
Tf~Weibull-Tr~Weibull	0.939	0.067	0.947	0.351397564

Porcentajes de cobertura para Intervalos de confianza del 90%

Tamaño de muestra n = 100

	W y K	p-valor	BS	p-valor
Tf-Normal-Tr-Normal	0.916	0.962	0.924	0.996240232
Tf-Normal-Tr-Exponencial	0.888	0.114	0.898	0.431800136
Tf-Normal-Tr-T Student	0.903	0.639	0.904	0.678445923
Tf-Normal-Tr-Lognormal	0.894	0.278	0.899	0.473400919
Tf-Normal-Tr-Gamma	0.906	0.751	0.913	0.924894164
Tf-Normal-Tr-Weibull	0.888	0.114	0.895	0.31377421
Tf-Exponencial-Tr-Normal	0.883	0.043 *	0.893	0.244272347
Tf-Exponencial-Tr-Exponencial	0.864	2E-04 *	0.880	0.021996145 *
Tf-Exponencial-Tr-T Student	0.898	0.432	0.906	0.751012182
Tf-Exponencial-Tr-Lognormal	0.892	0.213	0.900	0.51541771
Tf-Exponencial-Tr-Gamma	0.885	0.065	0.892	0.213040777
Tf-Exponencial-Tr-Weibull	0.898	0.432	0.899	0.473400919
Tf-T Student-Tr-Normal	0.911	0.889	0.915	0.951497493
Tf-T Student-Tr-Exponencial	0.907	0.784	0.914	0.939306269
Tf-T Student-Tr-T Student	0.903	0.639	0.906	0.751012182
Tf-T Student-Tr-Lognormal	0.886	0.079	0.896	0.351595097
Tf-T Student-Tr-Gamma	0.892	0.213	0.897	0.391060371
Tf-T Student-Tr-Weibull	0.905	0.716	0.914	0.939306269

	W y K	p-valor	BS	p-valor
Tf-lognormal-Tr-Normal	0.867	5E-04 *	0.876	0.007930712 *
Tf-lognormal-Tr-Exponencial	0.881	0.028 *	0.891	0.184347576
Tf-lognormal-Tr-T Student	0.897	0.391	0.896	0.351595097
Tf-lognormal-Tr-Lognormal	0.860	4E-05 *	0.879	0.017257231 *
Tf-lognormal-Tr-Gamma	0.867	5E-04 *	0.879	0.017257231 *
Tf-lognormal-Tr-Weibull	0.888	0.114	0.892	0.213040777
Tf-Gamma-Tr-Normal	0.887	0.095	0.903	0.639340372
Tf-Gamma-Tr-Exponencial	0.874	0.005 *	0.887	0.095408006
Tf-Gamma-Tr-T Student	0.899	0.473	0.910	0.866596228
Tf-Gamma-Tr-Lognormal	0.877	0.01 *	0.894	0.277914406
Tf-Gamma-Tr-Gamma	0.884	0.053	0.902	0.598846193
Tf-Gamma-Tr-Weibull	0.886	0.079	0.900	0.51541771
Tf-Weibull-Tr-Normal	0.884	0.053	0.905	0.715779842
Tf-Weibull-Tr-Exponencial	0.886	0.079	0.899	0.473400919
Tf-Weibull-Tr-T Student	0.894	0.278	0.906	0.751012182
Tf-Weibull-Tr-Lognormal	0.879	0.017 *	0.906	0.751012182
Tf-Weibull-Tr-Gamma	0.897	0.391	0.904	0.678445923
Tf-Weibull-Tr-Weibull	0.887	0.095	0.898	0.431800136

Porcentajes de cobertura para Intervalos de confianza del 95%

Tamaño de muestra n = 100

	W y K	p-valor	BS	p-valor
Tf-Normal-Tr-Normal	0.957	0.863	0.958	0.893703829
Tf-Normal-Tr-Exponencial	0.944	0.21	0.945	0.252881914
Tf-Normal-Tr-T Student	0.945	0.253	0.949	0.462470959
Tf-Normal-Tr-Lognormal	0.925	4E-04 *	0.931	0.005108727 *
Tf-Normal-Tr-Gamma	0.949	0.462	0.953	0.68846505
Tf-Normal-Tr-Weibull	0.947	0.351	0.952	0.634439985
Tf-Exponencial-Tr-Normal	0.948	0.406	0.949	0.462470959
Tf-Exponencial-Tr-Exponencial	0.926	6E-04 *	0.933	0.01059131 *
Tf-Exponencial-Tr-T Student	0.945	0.253	0.943	0.172057739
Tf-Exponencial-Tr-Lognormal	0.931	0.005 *	0.942	0.138918946
Tf-Exponencial-Tr-Gamma	0.940	0.087	0.949	0.462470959
Tf-Exponencial-Tr-Weibull	0.935	0.021 *	0.941	0.110561522
Tf-T Student-Tr-Normal	0.955	0.785	0.954	0.739035683
Tf-T Student-Tr-Exponencial	0.943	0.172	0.947	0.351397564
Tf-T Student-Tr-T Student	0.964	0.986	0.968	0.997804198
Tf-T Student-Tr-Lognormal	0.934	0.015 *	0.940	0.086732178
Tf-T Student-Tr-Gamma	0.958	0.894	0.963	0.978845168
Tf-T Student-Tr-Weibull	0.951	0.578	0.955	0.785317078

	W y K	p-valor	BS	p-valor
Tf-lognormal-Tr-Normal	0.948	0.406	0.949	0.462470959
Tf-lognormal-Tr-Exponencial	0.931	0.005 *	0.939	0.067062519
Tf-lognormal-Tr-T Student	0.935	0.021 *	0.944	0.210076079
Tf-lognormal-Tr-Lognormal	0.944	0.21	0.947	0.351397564
Tf-lognormal-Tr-Gamma	0.934	0.015 *	0.945	0.252881914
Tf-lognormal-Tr-Weibull	0.934	0.015 *	0.943	0.172057739
Tf-Gamma-Tr-Normal	0.947	0.351	0.951	0.577986161
Tf-Gamma-Tr-Exponencial	0.939	0.067	0.943	0.172057739
Tf-Gamma-Tr-T Student	0.951	0.578	0.954	0.739035683
Tf-Gamma-Tr-Lognormal	0.924	3E-04 *	0.937	0.038393236 *
Tf-Gamma-Tr-Gamma	0.945	0.253	0.949	0.462470959
Tf-Gamma-Tr-Weibull	0.948	0.406	0.951	0.577986161
Tf-Weibull-Tr-Normal	0.938	0.051	0.945	0.252881914
Tf-Weibull-Tr-Exponencial	0.933	0.011 *	0.938	0.051109559
Tf-Weibull-Tr-T Student	0.942	0.139	0.947	0.351397564
Tf-Weibull-Tr-Lognormal	0.931	0.005 *	0.936	0.028428397 *
Tf-Weibull-Tr-Gamma	0.943	0.172	0.948	0.405816073
Tf-Weibull-Tr-Weibull	0.938	0.051	0.940	0.086732178

Anexo D

Código del programa

Código del Algoritmo que genera los porcentajes de cobertura de los intervalos de confianza de la disponibilidad.

Librería Basics.h

```
/* librerías */
#include <math.h>

/* constantes para la Z */
#define Z_05 1.64485 /* Z_alfa/2 con 1-alfa = .90 */
#define Z_025 1.96 /* Z_alfa/2 con 1-alfa = .95 */

/* constantes para diversas funciones: dentro_IC, distribución Normal( 0, 1 ) */
#define FALSO 0
#define VERDADERO 1

/* constante para la función: MA_WEIBULL( *** ) */
#define PI 3.141592

/* función que regresa el cuadrado de un número */
float cuadrado( float x )
/* x es parámetro de entrada */
{
    return( x * x );
}

/* función que calcula la media aritmética de una muestra de tamaño n */
float media_aritmetica( float muestra[], int n )
/* muestra, n son parámetros de entrada */
{
    int i;
    float suma = 0.0;

    for ( i = 0; i < n; i++ )
        suma += muestra[i];

    return( suma / n );
}

/* función que calcula la varianza de una muestra de tamaño n */
float varianza( float muestra[], float media, int n, int bandera )
/* muestra, media, n, bandera son parámetros de entrada */
```

```

/* bandera = 0 --> varianza muestral */
/* bandera = 1 --> varianza poblacional */
{
    int i;
    float suma = 0.0;

    for ( i = 0; i < n; i++ )
        suma += cuadrado( muestra[i] - media );

    if ( bandera == 0 )
        return( suma / ( n - 1 ) );
    else
        return( suma / n );
}

/* función que calcula el tercer momento de una muestra de tamaño n */
float momento3( float muestra[], float media, int n )
/* muestra, media, n son parámetros de entrada */
{
    int i;
    float suma = 0.0;

    for ( i = 0; i < n; i++ )
        suma += pow( muestra[i] - media, 3 );

    return( suma / n );
}

/* función que calcula el valor de Z_alfa/2 para algunos valores de alfa */
float Z_alfa_en_2( int nivelDeConfianza )
/* nivelDeConfianza es parámetro de entrada */
/* nivelDeConfianza = 90, 95 */
{
    float Z;

    switch ( nivelDeConfianza )
    {
        case 90: Z = Z_05;           break;
        case 95: Z = Z_025;        break;
    }

    return( Z );
}

/* función que calcula la longitud media del IC */
float desplazamiento_IC( int nivelDeConfianza, float desviacionEstandarMuestral )
/* nivelDeConfianza, desviacionEstandarMuestral son parámetros de entrada */
{
    return( Z_alfa_en_2( nivelDeConfianza ) * desviacionEstandarMuestral );
}

/* función que indica si un valor determinado cae dentro del IC */
int dentro_IC( float valor, float limiteInferior, float limiteSuperior )
/* valor, limiteInferior, limiteSuperior son parámetros de entrada */
{

```

```

int dentro = FALSO;

if ( ( valor >= limiteInferior ) && ( valor <= limiteSuperior ) )
    dentro = VERDADERO;

return( dentro );
}

float funcion_gamma( float alfa )
{
    if ( ( alfa == 0.5 ) || ( alfa == 1.0 ) )
    {
        if ( alfa == 0.5 )
            return( sqrt( PI ) );
        else
            return( 1.0 );
    }
    else
        return( ( alfa - 1 ) * funcion_gamma( alfa - 1 ) );
}

```

Librería vac.h

```

/* librerías */
#include <stdlib.h>
#include "basics.h"

/* constantes para la distribución Uniforme( 0, 1 ) */
#define IM1    2147483563
#define IM2    2147483399
#define AM     ( 1.0 / IM1 )
#define IMM1   ( IM1 - 1 )
#define IA1    40014
#define IA2    40692
#define IQ1    53668
#define IQ2    52774
#define IR1    12211
#define IR2    3791
#define NTAB   32
#define NDIV   ( 1 + IMM1 / NTAB )
#define EPS    1.2e-7
#define RNMX   ( 1.0 - EPS )

/* función que genera un número aleatorio con distribución Uniforme( 0, 1 ) */
float UNIF_01( long *idum )
/* idum ( semilla ) es parámetro de salida */
{
    int j;
    long k;
    static long idum2 = 123456789;
    static long iy = 0;
    static long iv[NTAB];
    float temp;

```

```

if ( *idum <= 0 )
{
    if ( -( *idum ) < 1 )
        *idum = 1;
    else
        *idum = -( *idum );
    idum2 = ( *idum );
    for ( j = NTAB + 7; j >= 0; j-- )
    {
        k = ( *idum ) / IQ1;
        *idum = IA1 * ( *idum - k * IQ1 ) - k * IR1;
        if ( *idum < 0 )
            *idum += IM1;
        if ( j < NTAB )
            iv[j] = *idum;
    }
    iy = iv[0];
}
k = ( *idum ) / IQ1;
*idum = IA1 * ( *idum - k * IQ1 ) - k * IR1;
if ( *idum < 0 )
    *idum += IM1;
k = idum2 / IQ2;
idum2 = IA2 * ( idum2 - k * IQ2 ) - k * IR2;
if ( idum2 < 0 )
    idum2 += IM2;
j = iy / NDIV;
iy = iv[j] - idum2;
iv[j] = *idum;
if ( iy < 1 )
    iy += IMM1;
temp = AM * iy;

return( min( temp, RNMX ) );
}

/* función que genera un número aleatorio con distribución Normal(0,1) */
float NORMAL_01( long *semilla )
/* semilla es parámetro de salida */
{
    static int previo = FALSO;
    static float x;
    float y, w, v1, v2;

    if ( !previo )
    {
        do
        {
            v1 = 2.0 * UNIF_01( semilla ) - 1.0;
            v2 = 2.0 * UNIF_01( semilla ) - 1.0;
            w = cuadrado ( v1 ) + cuadrado( v2 );
        }
        while ( ( w >= 1.0 ) || ( w == 0.0 ) );
        y = sqrt( -2.0 * log( w ) / w );
        x = v1 * y;
        previo = VERDADERO;
    }
}

```

```

        return( v2 * y );
    }
    else
    {
        previo = FALSO;
        return( x );
    }
}

/* función que genera un número aleatorio con distribución Normal( M, S2 ) */
float NORMAL_M_S2( float media, float sigmaCuadrada, long *semilla )
/* media, sigmaCuadrada son parámetros de entrada */
/* semilla es parámetro de salida */
{
    return( media + sqrt( sigmaCuadrada ) * NORMAL_01( semilla ) );
}

/* función que genera un número aleatorio con distribución exponencial( beta ) */
float EXPONENCIAL( float beta, long *semilla )
/* beta es parámetro de entrada */
/* semilla es parámetro de salida */
{
    float x, u;

    u = UNIF_01( semilla );
    x = -beta * log( u );

    return( x );
}

/* función que genera un número aleatorio con distribución Lognormal( M, S2 ) */
float LOGNORMAL_M_S2( float media, float sigmaCuadrada, long *semilla )
/* media, sigmaCuadrada son parámetros de entrada */
/* semilla es parámetro de salida */
{
    return( exp ( NORMAL_M_S2( media, sigmaCuadrada, semilla ) ) );
}

/* función que genera un número aleatorio con distribución gamma( alfa < 1 ) */
float gamma_alfa_menor_1( float alfa, long *semilla )
/* alfa es parámetro de entrada */
/* semilla es parámetro de salida */
{
    static float b;
    float u1, u2, P, Y, comp;

    b = 1.0 + alfa / exp( 1 );
    do
    {
        u1 = UNIF_01( semilla );
        u2 = UNIF_01( semilla );
        P = b * u1;
        if ( P > 1 )
        {
            Y = -log( ( b - P ) / alfa );
            comp = pow( Y, alfa - 1 );
        }
    }
}

```

```

        else
        {
            Y = pow( P, 1 / alfa );
            comp = exp( -Y );
        }
    }
    while ( u2 > comp );

    return( Y );
}

/* función que genera un número aleatorio con distribución gamma( alfa > 1 ) */
float gamma_alfa_mayor_1( float alfa, long *semilla )
/* alfa es parámetro de entrada */
/* semilla es parámetro de salida */
{
    static float a, b, q, teta, d;
    float u1, u2, V, Y, Z, W, comp;

    a = 1.0 / sqrt( 2.0 * alfa - 1 );
    b = alfa - log( 4 );
    q = alfa + 1.0 / a;
    teta = 4.5;
    d = 1 + log( teta );

    do
    {
        u1 = UNIF_01( semilla );
        u2 = UNIF_01( semilla );
        V = a * log( u1 / ( 1 - u1 ) );
        Y = alfa * exp( V );
        Z = cuadrado( u1 ) * u2;
        W = b + q * V - Y;
        comp = W + d - teta * Z;
        if ( comp < 0 )
            comp = W - log( Z );
    }
    while ( comp < 0 );

    return( Y );
}

/* función que genera un número aleatorio con distribución gamma( alfa, 1 ) */
float GAMMA_A_1( float alfa, long *semilla )
/* alfa es parámetro de entrada */
/* semilla es parámetro de salida */
{
    if ( alfa == 1.0 )
        return( EXPONENCIAL( 1, semilla ) );
    else
        if ( alfa < 1.0 )
            return( gamma_alfa_menor_1( alfa, semilla ) );
        else
            return( gamma_alfa_mayor_1( alfa, semilla ) );
}

/* función que genera un número aleatorio con distribución gamma( alfa, beta ) */

```

```

float GAMMA_A_B( float alfa, float beta, long *semilla )
/* alfa, beta son parámetros de entrada */
/* semilla es parámetro de salida */
{
    return( beta * GAMMA_A_1( alfa, semilla ) );
}

/* función que genera un número aleatorio con distribución ji-cuadrada( grados ) */
float JI_CUADRADA( float grados, long *semilla )
/* grados es parámetro de entrada */
/* semilla es parámetro de salida */
{
    return( GAMMA_A_B( grados / 2, 2, semilla ) );
}

/* función que genera un número aleatorio con distribución t-student( grados ) */
float T_DE_STUDENT( float grados, long *semilla )
/* grados es parámetro de entrada */
/* semilla es parámetro de salida */
{
    return( NORMAL_01( semilla ) / sqrt( JI_CUADRADA( grados, semilla ) / grados ) );
}

/* función que genera un número aleatorio con distribución weibull( alfa, beta ) */
float WEIBULL( float alfa, float beta, long *semilla )
/* alfa, beta son parámetros de entrada */
/* semilla es parámetro de salida */
{
    float x, u;

    u = UNIF_01( semilla );
    x = beta * pow( - log( u ), 1 / alfa );

    return( x );
}

```

Librería mac.h

```

/* librerías */
#include "vac.h"

/* función que realiza una traslación y un escalamiento de una v.a.c. para que ésta tenga
media = mediaCero y sigma = sigmaCero */
float shift_scale( float vac, float mediaPrima, float sigmaPrima, float mediaCero, float
sigmaCero )
/* vac, mediaPrima, sigmaPrima, mediaCero, sigmaCero son parámetros de entrada */
{
    return( ( sigmaCero / sigmaPrima ) * ( vac - mediaPrima ) + mediaCero );
}

/* función que genera una muestra aleatoria de tamaño n con distribución Normal( M, S2 )
con media = mediaCero y sigma = sigmaCero */

```



```

void MA_NORMAL_M_S2( float media, float sigmaCuadrada, float n, float muestra[], long
*semilla )
/* media, sigmaCuadrada, n son parámetros de entrada */
/* muestra, semilla son parámetros de salida */
{
    int i;

    for ( i = 0; i < n; i++ )
        muestra[i] = NORMAL_M_S2( media, sigmaCuadrada, semilla );
}

/* función que genera una muestra aleatoria de tamaño n con distribución exponencial( beta
) con media = mediaCero y sigma = sigmaCero */
void MA_EXPONENCIAL( float beta, float n, float muestra[], long *semilla, float mediaCero,
float sigmaCero )
/* grados, n, mediaCero, sigmaCero son parámetros de entrada */
/* muestra, semilla son parámetros de salida */
{
    int i;
    float Y;

    for ( i = 0; i < n; i++ )
    {
        Y = EXPONENCIAL( beta, semilla );
        muestra[i] = shift_scale( Y, beta, beta, mediaCero, sigmaCero );
    }
}

/* función que genera una muestra aleatoria de tamaño n con distribución t-student( grados )
con media = mediaCero y sigma = sigmaCero */
void MA_T_DE_STUDENT( float grados, float n, float muestra[], long *semilla, float
mediaCero, float sigmaCero )
/* grados, n, mediaCero, sigmaCero son parámetros de entrada */
/* muestra, semilla son parámetros de salida */
{
    int i;
    float Y;

    for ( i = 0; i < n; i++ )
    {
        Y = T_DE_STUDENT( grados, semilla );
        muestra[i] = shift_scale( Y, 0, sqrt( grados / ( grados - 2 ) ), mediaCero, sigmaCero
);
    }
}

/* función que genera una muestra aleatoria de tamaño n con distribución Lognormal( M, S2
) con media = mediaCero y sigma = sigmaCero */
void MA_LOGNORMAL_M_S2( float media, float sigmaCuadrada, float n, float muestra[],
long *semilla, float mediaCero, float sigmaCero )
/* media, sigmaCuadrada, n, mediaCero, sigmaCero son parámetros de entrada */
/* muestra, semilla son parámetros de salida */
{
    int i;
    float Y;

    for ( i = 0; i < n; i++ )

```

```

    {
        Y = LOGNORMAL_M_S2( media, sigmaCuadrada, semilla );
        muestra[i] = shift_scale( Y, exp( media + sigmaCuadrada / 2 ), sqrt( exp( 2 * media
+ sigmaCuadrada ) * ( exp( sigmaCuadrada ) - 1 ) ), mediaCero, sigmaCero );
    }
}

/* función que genera una muestra aleatoria de tamaño n con distribución gamma( alfa, beta
) con media = mediaCero y sigma = sigmaCero */
void MA_GAMMA_A_B( float alfa, float beta, float n, float muestra[], long *semilla, float
mediaCero, float sigmaCero )
/* alfa, beta, n, mediaCero, sigmaCero son parámetros de entrada */
/* muestra, semilla son parámetros de salida */
{
    int i;
    float Y;

    for ( i = 0; i < n; i++ )
    {
        Y = GAMMA_A_B( alfa, beta, semilla );
        muestra[i] = shift_scale( Y, alfa * beta, sqrt( alfa * cuadrado( beta ) ), mediaCero,
sigmaCero );
    }
}

/* función que genera una muestra aleatoria de tamaño n con distribución weibull( alfa, beta
) con media = mediaCero y sigma = sigmaCero */
void MA_WEIBULL( float alfa, float beta, float n, float muestra[], long *semilla, float
mediaCero, float sigmaCero )
/* alfa, beta, n, mediaCero, sigmaCero son parámetros de entrada */
/* muestra, semilla son parámetros de salida */
{
    int i;
    float Y;

    for ( i = 0; i < n; i++ )
    {
        Y = WEIBULL( alfa, beta, semilla );
        muestra[i] = shift_scale( Y, ( beta / alfa ) * funcion_gamma( 1 / alfa ) , sqrt( (
cuadrado( beta ) / alfa ) * ( 2 * funcion_gamma( 2 / alfa ) - ( 1 / alfa ) * cuadrado(
funcion_gamma( 1 / alfa ) ) ) ), mediaCero, sigmaCero );
    }
}

```

Programa sem.c

```

#include <stdio.h>
#include <time.h>
#include "vac.h"

void main()
{
    int i;
    long semilla;
    //long semilla = -278577;

```

```

randomize();
semilla = (long) ( -1 * rand() );
for ( i = 0; i < 20; i++ )
    printf( "%f\n", UNIF_01( &semilla ) );
}

```

Programa btrap.c

```

/* librerías */
#include <stdio.h>
#include "mac.h"

/* constantes para las muestras */
#define MAXMUESTRA100 /* tamaño máximo de la muestra */
#define NUM_DISTR      6 /* número de distribuciones */
#define NUM_IC         2 /* número de intervalos de confianza */
#define NUM_N0         4 /* número de n0's */

/* constantes relacionadas con el proceso principal: A's, método bootstrap, IC's */
#define M_TF           55.1 /* media real de los tiempos de falla */
#define VARIANZA_TF    2747.49 /* varianza real de los tiempos de falla */
#define SIGMA_TF       52.4165 /* desviación estándar real de los tiempos
de falla */
#define M_TR           2.49 /* media real de los tiempos de reparación */
#define VARIANZA_TR    7.7919 /* varianza real de los tiempos de reparación */
#define SIGMA_TR       2.7913 /* desviación estándar real de los tiempos de
reparación */
#define B              1000 /* número de muestras bootstrap */
#define NUM_ITER       1000 /* número de iteraciones para crear los IC's */
#define CONFIANZA_BI_1 90 /* nivel de confianza para IC bilateral al 90% */
#define CONFIANZA_BI_2 95 /* nivel de confianza para IC bilateral al 95% */

/* función que calcula la disponibilidad */
float disponibilidad( float tf, float tr )
/* tf, tr son parámetros de entrada */
{
    return( tf / ( tf + tr ) );
}

/* función que genera una muestra aleatoria bootstrap */
void muestra_aleatoria_bootstrap( float muestraFija[], int n, float muestraVariante[], long
*semilla )
/* muestraFija, n son parámetros de entrada */
/* muestraVariante, semilla son parámetros de salida */
{
    int i;

    for ( i = 0; i < n; i++ )
        muestraVariante[i] = muestraFija[( int ) ( UNIF_01( semilla ) * n )];
}

/* función que crea A(i) para i = 1, ..., B */

```

```

void distribucion_disponibilidad_bootstrap( float muestraFijaTF[], float muestraFijaTR[], int n,
float ADistr[], long *semilla )
/* muestraFijaTF, muestraFijaTR, n son parámetros de entrada */
/* ADistr, semilla son parámetros de salida */
{
    int i;
    float muestraVariante[MAXMUESTRA], mediaAritmeticaTF, mediaAritmeticaTR;

    for ( i = 0; i < B; i++ )
    {
        muestra_aleatoria_bootstrap( muestraFijaTF, n, muestraVariante, semilla );
        mediaAritmeticaTF = media_aritmetica( muestraVariante, n );

        muestra_aleatoria_bootstrap( muestraFijaTR, n, muestraVariante, semilla );
        mediaAritmeticaTR = media_aritmetica( muestraVariante, n );

        ADistr[i] = disponibilidad( mediaAritmeticaTF, mediaAritmeticaTR );
    }
}

/* función principal */
void main()
{
    long semilla = -278577; /* semilla inicial para el generador de números aleatorios uniformes */
    float N[NUM_N0] = { 20, 30, 50, 100 }; /* valores para nCero */
    float muestraTF[MAXMUESTRA]; /* muestra generada para los tiempos de falla */
    float muestraTR[MAXMUESTRA]; /* muestra generada para los tiempos de reparación */
    int nCero; /* tamaño de la muestra = 20, 30, 50, 100 */
    float AReal; /* A obtenida a partir de los parámetros reales */
    float mediaMuestralTF; /* parámetro de la muestra generada para los tiempos de falla */
    float mediaMuestralTR; /* parámetro de la muestra generada para los tiempos de
reparación */
    float varianzaMuestralTF; /* parámetro de la muestra generada para los tiempos de falla */
    float varianzaMuestralTR; /* parámetro de la muestra generada para los tiempos de
reparación */
    float AIntervalo; /* A que se incluye en los IC's (SB y W&K) */
    float deltaAEstimada, varianzaMuestralWK, desviacionEstandarMuestralWK; /* parámetros
del método W&K */
    float varianzaMuestralWK_termino1, varianzaMuestralWK_termino2; /* parámetros
auxiliares de la varianzaWK */
    float *As; /* distribución de las disponibilidades 1 -> B para SB */
    float AAsterisco; /* parámetros de los índices bootstrap */
    float SAAsterisco; /* parámetros de los índices bootstrap */
    float desplazamientoICA_SB, desplazamientoICA_WK; /* variables auxiliares para los IC's
(SB y W&K) */
    int conta_SB, conta_WK; /* contador de las veces que la A real cae dentro de los IC's
obtenidos (SB y W&K) */
    float porcentajeA_SB, porcentajeA_WK; /* porcentajes de cobertura para los IC's (SB y
W&K) */
    int distribucionTF, distribucionTR, tipoIntervalo, i, j; /* 1 -> NUM_DISTR, 1 -> NUM_DISTR,
NUM_IC, NUM_N0, NUM_ITER respectivamente */
    FILE *archivo; /* apuntador a un archivo de texto */
    char *nombreArchivo[NUM_DISTR][NUM_DISTR][NUM_IC] =
    {
        {
            { "90N_N.txt", "95N_N.txt" },

```

```

        { "90N_EX.txt", "95N_EX.txt" },
        { "90N_T.txt", "95N_T.txt" },
        { "90N_LN.txt", "95N_LN.txt" },
        { "90N_GA.txt", "95N_GA.txt" },
        { "90N_W.txt", "95N_W.txt" }
    },
    {
        { "90EX_N.txt", "95EX_N.txt" },
        { "90EX_EX.txt", "95EX_EX.txt" },
        { "90EX_T.txt", "95EX_T.txt" },
        { "90EX_LN.txt", "95EX_LN.txt" },
        { "90EX_GA.txt", "95EX_GA.txt" },
        { "90EX_W.txt", "95EX_W.txt" }
    },
    {
        { "90T_N.txt", "95T_N.txt" },
        { "90T_EX.txt", "95T_EX.txt" },
        { "90T_T.txt", "95T_T.txt" },
        { "90T_LN.txt", "95T_LN.txt" },
        { "90T_GA.txt", "95T_GA.txt" },
        { "90T_W.txt", "95T_W.txt" }
    },
    {
        { "90LN_N.txt", "95LN_N.txt" },
        { "90LN_EX.txt", "95LN_EX.txt" },
        { "90LN_T.txt", "95LN_T.txt" },
        { "90LN_LN.txt", "95LN_LN.txt" },
        { "90LN_GA.txt", "95LN_GA.txt" },
        { "90LN_W.txt", "95LN_W.txt" }
    },
    {
        { "90GA_N.txt", "95GA_N.txt" },
        { "90GA_EX.txt", "95GA_EX.txt" },
        { "90GA_T.txt", "95GA_T.txt" },
        { "90GA_LN.txt", "95GA_LN.txt" },
        { "90GA_GA.txt", "95GA_GA.txt" },
        { "90GA_W.txt", "95GA_W.txt" }
    },
    {
        { "90W_N.txt", "95W_N.txt" },
        { "90W_EX.txt", "95W_EX.txt" },
        { "90W_T.txt", "95W_T.txt" },
        { "90W_LN.txt", "95W_LN.txt" },
        { "90W_GA.txt", "95W_GA.txt" },
        { "90W_W.txt", "95W_W.txt" }
    }
}
}; /* nombre del archivo de texto de salida */

As = (float *) malloc( B * sizeof( float ) );

printf( "Empezó el proceso\n\n" );
AReal = disponibilidad( M_TF, M_TR );
for ( distribucionTF = 0; distribucionTF < NUM_DISTR; distribucionTF++ )
    for ( distribucionTR = 0; distribucionTR < NUM_DISTR; distribucionTR++ )
        {
            for ( tipoIntervalo = 0; tipoIntervalo < NUM_IC; tipoIntervalo++ )
                {

```

```

        archivo = fopen(
nombreArchivo[distribucionTF][distribucionTR][tipoIntervalo], "w" );
        for ( i = 0; i < NUM_N0; i++ )
        {
            nCero = N[i];

            contA_SB = 0;
            contA_WK = 0;

            for ( j = 0; j < NUM_ITER; j++ )
            {
                switch ( distribucionTF )
                {
                    case 0: MA_NORMAL_M_S2( M_TF, VARIANZA_TF,
nCero, muestraTF, &semilla );
                                break;
                    case 1: MA_EXPONENCIAL( 1, nCero, muestraTF,
&semilla, M_TF, SIGMA_TF );
                                break;
                    case 2: MA_T_DE_STUDENT( 4, nCero, muestraTF,
&semilla, M_TF, SIGMA_TF );
                                break;
                    case 3: MA_LOGNORMAL_M_S2( 0, 1, nCero, muestraTF,
&semilla, M_TF, SIGMA_TF );
                                break;
                    case 4: MA_GAMMA_A_B( 2, 1, nCero, muestraTF,
&semilla, M_TF, SIGMA_TF );
                                break;
                    case 5: MA_WEIBULL( 2, 1, nCero, muestraTF, &semilla,
M_TF, SIGMA_TF );
                                break;
                }

                switch ( distribucionTR )
                {
                    case 0: MA_NORMAL_M_S2( M_TR, VARIANZA_TR,
nCero, muestraTR, &semilla );
                                break;
                    case 1: MA_EXPONENCIAL( 1, nCero, muestraTR,
&semilla, M_TR, SIGMA_TR );
                                break;
                    case 2: MA_T_DE_STUDENT( 4, nCero, muestraTR,
&semilla, M_TR, SIGMA_TR );
                                break;
                    case 3: MA_LOGNORMAL_M_S2( 0, 1, nCero,
muestraTR, &semilla, M_TR, SIGMA_TR );
                                break;
                    case 4: MA_GAMMA_A_B( 2, 1, nCero, muestraTR,
&semilla, M_TR, SIGMA_TR );
                                break;
                    case 5: MA_WEIBULL( 2, 1, nCero, muestraTR, &semilla,
M_TR, SIGMA_TR );
                                break;
                }

                // método W&K
                mediaMuestralTF = media_aritmetica( muestraTF, nCero );
            }
        }

```

```

varianzaMuestralTF = varianza( muestraTF,
mediaMuestralTF, nCero, 1 );

mediaMuestralTR = media_aritmetica( muestraTR, nCero );
varianzaMuestralTR = varianza( muestraTR,
mediaMuestralTR, nCero, 1 );

AIntervalo = disponibilidad( mediaMuestralTF,
mediaMuestralTR );

deltaAEstimada = ( mediaMuestralTF * varianzaMuestralTR
- mediaMuestralTR * varianzaMuestralTF ) / ( nCero * pow ( mediaMuestralTF +
mediaMuestralTR, 3 ) );

varianzaMuestralWK_termino1 = ( cuadrado(
mediaMuestralTR ) * varianzaMuestralTF + cuadrado( mediaMuestralTF ) *
varianzaMuestralTR ) / ( nCero * pow ( mediaMuestralTF + mediaMuestralTR, 4 ) );
varianzaMuestralWK_termino2 = ( 2 * cuadrado(
mediaMuestralTR ) * momento3( muestraTF, mediaMuestralTF, nCero ) + 2 * cuadrado(
mediaMuestralTF ) * momento3( muestraTR, mediaMuestralTR, nCero ) ) / ( cuadrado(
nCero ) * pow ( mediaMuestralTF + mediaMuestralTR, 5 ) );
varianzaMuestralWK = varianzaMuestralWK_termino1 -
varianzaMuestralWK_termino2;
desviacionEstandarMuestralWK = sqrt( varianzaMuestralWK
);

// método SB
distribucion_disponibilidad_bootstrap( muestraTF,
muestraTR, nCero, As, &semilla );
AAsterisco = media_aritmetica( As, B );
SAAsterisco = sqrt( varianza( As, AAsterisco, B, 0 ) );

if ( tipoIntervalo == 0 ) /* tipoIntervalo = 0 --> IC al 90% */
{
// IC's al 90% (SB y W&K)
desplazamientoICA_WK = desplazamiento_IC(
CONFIANZA_BI_1, desviacionEstandarMuestralWK );
desplazamientoICA_SB = desplazamiento_IC(
CONFIANZA_BI_1, SAAsterisco );
}
else /* tipoIntervalo = 1 --> IC al 95% */
{
// IC's al 95% (SB y W&K)
desplazamientoICA_WK = desplazamiento_IC(
CONFIANZA_BI_2, desviacionEstandarMuestralWK );
desplazamientoICA_SB = desplazamiento_IC(
CONFIANZA_BI_2, SAAsterisco );
}

if ( dentro_IC( AReal, AIntervalo - deltaAEstimada -
desplazamientoICA_WK, AIntervalo - deltaAEstimada + desplazamientoICA_WK ) )
contA_WK++;

if ( dentro_IC( AReal, AIntervalo - desplazamientoICA_SB,
AIntervalo + desplazamientoICA_SB ) )
contA_SB++;
} // termina ciclo de N

```

```

fprintf( archivo, "n = %d\n\n", nCero );

porcentajeA_WK = ( 1.0 * contA_WK ) / NUM_ITER;
porcentajeA_SB = ( 1.0 * contA_SB ) / NUM_ITER;

fprintf( archivo, "%.3f\n\n", porcentajeA_WK );
fprintf( archivo, "%.3f\n\n", porcentajeA_SB );
} // termina ciclo de nCero
fclose( archivo );
printf( "Terminó la tabla con distribuciónTF = %d, distribuciónTR = %d, IC =
%d. Total: %d / %d\n", distribucionTF, distribucionTR, tipoIntervalo, ( distribucionTF *
NUM_DISTR * NUM_IC ) + ( distribucionTR * NUM_IC ) + ( tipoIntervalo + 1 ), NUM_DISTR
* NUM_DISTR * NUM_IC );
} // termina ciclo de IC
} // termina ciclo de distribuciónTR
} // fin de la función main()

```


Referencias bibliográficas

CASELLA, George y Berger, Roger L, (1990). Statistical Inference. Duxbury Press, Estados Unidos.

EFRON, Bradley (1979). Bootstrap methods: Another look at the jackknife. The Annals of statistics, 7, 1-2.

EFRON, Bradley (1982). The Jackknife, the Bootstrap and other resampling plans. Philadelphia: SIAM.

EFRON, Bradley y Tibshirany, Robert J. (1986). Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. Statistical Science 1 (1), 54-77.

EFRON, Bradley (1987). "Better Bootstrap Confidence Intervals". Journal of the American Statistical Association, Vol. 82, No. 397, pag. 171-200, Marzo 1987.

EFRON, Bradley y Tibshirany, Robert J. (1993). An Introduction to the Bootstrap. Chapman & Hall, Estados Unidos.

GOTTFRIED, Byron S. (1997). Programación en C. Segunda edición, McGraw-Hill Interamericana de España, España.

HINCKLEY, D. W. (1988). Bootstrap Methods. Journal of the royal Statistical society, series B, 50, 321-337.

IRESON, W. Grant (1966), Reliability Handbook, McGraw-Hill Inc. Estados Unidos.

KECECIOGLU, B. Dimitri (1993), Reliability & life testing. Third printing 1997, Prentice-Hall, Inc., New Jersey.

KECECIOGLU, B. Dimitri (1995), Maintainability, Availability Operational Readiness Engineering, Prentice-Hall, Inc., New Jersey.

LAW, Averill M. y Kelton W. David (1991). Simulation Modeling and Analysis. Segunda edición, McGraw-Hill Inc., Estados Unidos

LAWLESS, J.F. (1982), Statistical Models and Methods for lifetime data. Wiley.

LEWIS E. E. (1987), Introduction to reliability engineering, Wiley, Estados Unidos

LUNNEBORG, C. E. (1987). Bootstrap Applications for the Behavioral Sciences: University of Washington.

MOONEY, C. Z, y Duval, R. D. (1993). Bootstrapping: A Nonparametric Approach to Statistical Inference. Newbury Park, CA: Sage.

RAMAKUMAR, R, (1993) Engineering Reliability: Fundamentals and Applications, Prentice Hall, Estados Unidos

WANG Wendai y Kececioglu Dimitri B. (2000) Confidence Limits on the inherent Availability of Equipment. AlliedSignal, Phoenix y The University of Arizona, Tucson. Proceedings Annual Reliability and Maintainability Symposium (IEEE Reliability Society)

Centro de Información-Biblioteca



30002006141246