Econometria com aplicações em R e C

Daniel Francisco Neyra Castañeda

2015

A mis padres Segundo y Maria.

Sumário

1 Introdução **1**

1.1 Como funciona a Econometria 1
1.2 Tipos de dados 1
1.3 Metodologia 2
1.3.1 Hipótese 2
1.3.2 Modelo matemático 2
1.3.3 Modelo econométrico 3
1.3.4 Nível de significância 3
1.3.5 Obtenção de dados 3
1.3.6 Estatística de prova 4
1.3.7 Tipos de erro 4
1.3.8 Regra de decisão 4
1.3.9 Conclusões 5
1.3.10 Previsões 5
1.3.11 Resíduos 5
1.4 Exercícios 6

2 Alguns comandos do R **7**

2.1 Iniciando o R 7
2.2 Dados permanentes e removendo objetos 7
2.3 Manipulação de números e vetores 8
2.4 Funções matemáticas 10
2.5 Funções estatísticas 10
2.6 Cores 11
2.7 Fatores ordenados e desordenados 12
2.8 Vetores e Matrizes 13
2.8.1 Indexando matrizes 13
2.8.2 Sequências 14
2.8.3 Multiplicação e inversão de matrizes 15
2.8.4 Comandos cbind() e rbind() 17
2.8.5 Autovalores e autovetores 17
2.9 Data frame ou construtor de dados 18
2.10 Funções attach() e detach() 18
2.11 Função read.table() 19

2.12 Script com extensão .R 20
2.13 Função scan() 20
2.14 Função sample 21
2.15 Rcmdr 21
2.16 Dados disponíveis no R 21
2.17 Modificando e editando dados 22
2.18 Exportando dados do R para o Excel 22
2.19 Funções: apply, tapply, sapply e lapply 22
2.20 Distribuições de probabilidade 23
2.21 Algoritmos de programação no R 24
2.22 Laços no R 26
2.23 A função cut() 26
2.24 Criando a suas próprias funções 27
2.24.1 Teste que compara duas médias 27
2.24.2 Coeficiente de regressão 27
2.25 Teste Qui- Quadrado 28
2.26 Teste t - student 29
2.27 Teste F 30
2.28 Gráficos 32
2.29 Tabelas 35
2.30 Integração por simulação 35
2.31 Comandos adicionais 36
2.32 Exercicios 36

3 Regressão linear simples **41**
3.1 Objetivo 41
3.2 Definição 41
3.3 Mínimos Quadrados Ordinários 43
3.4 Máxima Verosimilhança 43
3.5 Teorema de Gauss - Markov 44
3.6 Covariância de β 44
3.7 Previsões 44
3.8 Correlação 45
3.8.1 Teste de correlação 46
3.9 Coeficiente de determinação 47
3.10 Estimação de σ^2 47
3.11 Intervalo de confiança 48
3.12 Análise de variância 48
3.13 Regressão com erros normais 49
3.13.1 O modelo 49
3.13.2 A função `lm()` 49
3.14 Exemplo do valor pago 53
3.14.1 O modelo 54
3.14.2 Resumo dos resultados 54
3.15 Regressão linear nos parâmetros 55
3.15.1 Os gráficos 55

3.15.2 ANOVA 56
3.15.3 Hipóteses do modelo 56
3.15.4 As previsões 57
3.15.5 Os resíduos 57
3.15.6 Componentes adicionais da função lm 58
3.16 O modelo quadrático 58
3.16.1 Modelo matemático 59
3.16.2 Implementação no R 59
3.17 Modelos de potência 60
3.18 Comparando modelos 60
3.19 Derivação dos EMQO 61
3.20 Extensão da Regressão Linear Simples 61
3.20.1 Matriz de dispersão 63
3.20.2 ANOVA 63
3.21 Critério de Seleção de modelos 63
3.22 Exemplo 3 63
3.22.1 O modelo 64
3.22.2 As hipóteses 64
3.22.3 Estimação dos parâmetros 64
3.23 Modelo linear parcial 66
3.24 Seleção de regressores 67
3.25 Significância e amostra 69
3.26 Especificação do modelo 70
3.26.1 Teste de especificação Reset de Ramsey 70
3.26.2 Normalidade dos resíduos 70
3.27 Gráficos 71
3.28 Teste de mudança estrutural 71
3.29 Teste de outliers 72
3.30 Detecção de observações atípicas 72
3.31 Consistência 75
3.32 Iterações 76
3.33 Regressão separada 77
3.34 Regressão quantílica 78
3.35 Derivação de MQO na regressão Múltipla 79
3.36 Violação de um pressuposto básico 80
3.37 Aleatoriedade 81
3.38 Gerador de Números Aleatórios 81
3.39 Testes de Normalidade 82
3.39.1 Teste Jarque-Bera (JB) 83
3.39.2 Teste Lilliefors 84
3.40 Simulação de Monte Carlo 85
3.40.1 Implementação no R 85
3.40.2 Extensão para regressão múltipla 87
3.41 Exercícios 89

4 Casos especiais de Regressão **93**
4.1 ANOVA em delineamentos . . . 93
4.1.1 Inteiramente causalizado . . . 93
4.1.2 Bloco causalizado . . . 94
4.2 Modelos de efeitos fixos e aleatórios . . . 95
4.3 Simulando dados de regressão . . . 96

5 Programação em C **101**
5.1 Simulação em C . . . 101
5.2 Simulação de números aleatórios . . . 102
5.3 A simulação e o teste de Jarque-Bera . . . 102
5.4 Métodos . . . 102
5.5 Resultados . . . 103
5.6 Código C: Teste de normalidade de Jarque-Bera . . . 107
5.7 Código C: Poder do teste de normalidade de JB . . . 114
5.8 Código C: Teste de normalidade de Lilliefors . . . 115
5.9 Código C: Jarque-Bera: Modelo heterocedástico . . . 120

6 Superando os pressupostos básicos **121**
6.1 Heterocedasticidade . . . 121
6.1.1 Testes de heterocedasticidade . . . 124
6.1.2 Corrigindo a heterocedasticidade . . . 127
6.1.3 Estimador HC0 de White . . . 129
6.1.4 Estimador HC1 de Hinkler . . . 130
6.1.5 Estimador HC2 de Horn-Duncan . . . 130
6.1.6 Estimador HC3 de Davidson-MacKinnon . . . 130
6.1.7 Estimador HC4 de Cribari Neto . . . 130
6.1.8 Estimador HC5 de Cribari-Souza . . . 131
6.2 Multicolinearidade . . . 131
6.2.1 Natureza . . . 132
6.2.2 Identificação . . . 133
6.2.3 Corrigindo . . . 133
6.2.4 Testes de multicolinearidade . . . 133
6.2.5 Exemplo . . . 134
6.2.6 Regressão ridge . . . 135
6.3 Autocorrelação . . . 136
6.3.1 Natureza da autocorrelação . . . 137
6.3.2 Diagnostico da autocorrelação . . . 138
6.3.3 Eliminação da autocorrelação . . . 140
6.4 Exercícios . . . 141

7 Estimadores **143**
7.1 Propriedades de um estimador . . . 143
7.2 Algoritmos . . . 144
7.2.1 Viés . . . 144
7.2.2 Consistência . . . 145

7.2.3 EQM 145

8 Regressão não linear **147**
8.1 Otimização não linear 148
8.1.1 Steepest Ascend 149
8.1.2 Newton Rapshon 149
8.1.3 Método de Newton 149
8.1.4 Scoring de Fisher 149
8.1.5 Método Quase- Newton 149
8.1.6 Quase Newton 1 ou DFP 150
8.1.7 Método BFGS 150
8.2 Modelo exponencial 153
8.2.1 Medida de ajuste 153
8.2.2 Exemplo 154
8.3 Modelo logístico 154
8.4 Modelo alternativo 156
8.5 Regressão sobre variáveis dummies 158
8.5.1 Variáveis Dummies 158
8.5.2 Regressão sobre variáveis qualitativas e quantitativas . . . 160
8.6 Modelos de regressão para respostas binarias 162
8.7 Modelos de probabilidade linear 163
8.7.1 Modelo Probit 163
8.7.2 Modelo Logit 163
8.8 Modelo Weibull 163
8.9 Modelo C Log-log 163
8.9.1 Exemplo da razão de chances 164
8.9.2 Exemplo da qualidade dos funcionários 167
8.9.3 Teste de Hosmer Lemeshow 168
8.10 Exercícios 169

9 Séries temporais **171**
9.1 Introdução 171
9.2 Pacotes e funções para série temporais no R 171
9.3 Fontes 172
9.4 Regressão linear com séries temporais 175
9.5 Deflacionando uma série 178
9.6 Decomposição de uma série 179
9.6.1 Retornos 179
9.6.2 Decomposição clássica 180
9.6.3 Autocorrelação 182
9.6.4 O Correlograma 183
9.6.5 Processos estacionários 184
9.6.6 Passeio Aleatório 185
9.7 Alisamento exponencial 185
9.7.1 Algoritmo simples 185
9.7.2 Algoritmo de Holt 185

9.7.3 Algorimo de Holt-Winters(HW) . . . 186
9.7.4 HW multiplicativo . . . 186
9.8 ARMA e ARIMA . . . 187
9.8.1 Processos autorregressivos . . . 188
9.8.2 Processo de médias móveis . . . 188
9.8.3 Processo ARMA . . . 189
9.8.4 Processo ARIMA(p, d, q) . . . 190
9.8.5 Simulações de modelos Arima . . . 194
9.9 Modelos Sarima . . . 194
9.9.1 Analisando a Bovespa . . . 195
9.9.2 Analisando o IGP-DI . . . 196
9.9.3 Deflacionando o ICMS - SE . . . 198
9.9.4 Arima vs Sarima . . . 199
9.10 Processos ARFIMA . . . 200
9.10.1 Estimação pelo periodograma (GPH) . . . 202
9.10.2 Estimação pelo periodograma suavizado (GPH_S) . . . 202
9.10.3 Estimação pela máxima verossimilhança. . . . 203
9.11 Exercícios . . . 208
9.12 Modelos não lineares estruturais . . . 210
9.12.1 Teste de Ljung e Box . . . 210
9.12.2 ARCH . . . 210
9.12.3 Simulando ARCH . . . 213
9.12.4 GARCH . . . 213
9.12.5 Simulando GARCH . . . 218
9.13 Raízes unitárias . . . 219
9.14 Desempenho de modelos AR e MA no R . . . 223
9.14.1 Classificação . . . 224
9.14.2 Os cenários de simulação . . . 224
9.14.3 O algoritmo . . . 224
9.14.4 Os resultados . . . 226

10 Geradores de números aleatórios: Testes DIEHARD 229
10.1 Geração de números aleatórios . . . 230
10.2 Testes estatísticos . . . 230
10.3 O Sistema DIEHARD . . . 230
10.4 Material e métodos . . . 234
10.5 Resultados . . . 234
10.6 Conclusões . . . 235
10.7 Algoritmo em C . . . 237
10.7.1 Código C que implementa o teste da distância mínima. . . 237
10.7.2 Distância mínima no quadrado usando Marsaglia . . . 244
10.7.3 Esfera mínima usando o gerador de Marsaglia . . . 245
10.7.4 Distância mínima com os 640.000 primeiros dígitos de π . 246
10.7.5 Esfera mínima com os 640.000 primeiros dígitos de π . . . 246

Agradecimentos

Este livro é fruto das notas de aula da disciplina de Análise de regressão e Métodos Quantitativos em Econometria ministradas nos cursos de Estatística e Ciências Atuariais na Universidade Federal de Sergipe durante os últimos anos. Uma contribuição adicional a este livro são as disciplinas de Estatística Aplicada e de Séries Temporais, cursadas na pôs-graduação da UFPE em 2008. Especial agradecimento ao professor Francisco Cribari Neto (UFPE), por ter norteado as disciplinas que atualmente ministro, de igual forma estes agradecimentos são estendidos ao professor Marcelo Gama da Silva (UFRPE), colega e integrante do referido curso quem colaborou na construção dos algoritmos de programação na plataforma C.

O curso de Estatística na Universidade Federal de Sergipe, oferece atualmente uma disciplina de análise de Regressão e duas disciplinas de Econometria, e tem-se abordado modelos econométricos em algumas plataformas, tendo definido como padrão o **projeto R**, que além de ser livre e gratuita, utiliza a linguagem S, as suas funções já estão implementadas nos pacotes padrão ou pacotes externos a serem instalados, os quais sempre estão em constante atualização. O projeto R é uma linguagem de programação matemática orientada a objetos que usa comandos de linha para sua execução. Este ambiente vem imprimindo um novo ritmo de procura na comunidade universitária pela qualidade dos seus resultados, o tempo de execução é imediata, e pela quantidade de informação como material bibliográfico e referencias na internet no sitio: `www.r-project.org`. Com este trabalho tenta-se contornar a necessidade de aplicar os métodos estatísticos na medida econômica usando como plataforma padrão o **projeto R**, e em alguns casos com a programação em C.

Prefácio

No ambiente de trabalho da estatística aplicada não há mais espaço para perder tempo em cálculos matemáticos que são repetitivos, demorados ou em alguns casos sem solução analítica, por isso atualmente existe uma demanda crescente em calcular medidas ou técnicas estatísticas com algum software disponível no mercado com o objetivo de diminuir substancialmente o tempo para a realização desta tarefa. Muitos programas de computador auxiliam as análises de dados de pesquisas, estas análises recaem em pacotes estatísticos que no mercado são pagos, ou alugados com licenças que expiram periodicamente. No entanto a plataforma R, supera estes impasses, proporcionando um programa livre (o seu código fonte esta disponível para alteração pelo usuário) e gratuito (sem custo monetário) usando a linguagem de programação S, inicialmente difundida pelo SPLUS, cujo formato de trabalho é via comandos de linha. Este programa é confiável pela qualidade dos seus resultados, e pela estabilidade em ambientes como Linux e Windows, usaremos este último ambiente neste livro. Para trabalhar na plataforma R, bibliotecas externas podem ser instaladas, e para maior praticidade alguns destes pacotes usa ambiente de janelas, como exemplo o pacote `Rcmdr`. Atualmente o R disponibiliza mais de 1000 pacotes para serem instaladas na sua biblioteca, porem cada área utiliza funções específicas, por exemplo, em econometria, podemos estar interessados inicialmente nos pacotes: `lmtest`, `quantreg`, `KernSmooth`, `car`, `strucchange`, `AER`, além do pacote padrão `stats`. Para séries temporais, podemos acrescentar os pacotes: `tseries`, `forecast`, `fracdiff`, `vars`, todos estes pacotes podem ser facilmente baixados (download) via internet, além disso os seus tutoriais no formato htlm, pdf, ou ser procurados na ajuda (help) do R. Uma outra linguagem como o C é proposta para construção de algoritmo e programas no decorrer do livro. Esta linguagem de programação de aplicações é a mais comum no sistema operativo Windows. Nesta primeira edição, o objetivo inicial, é apresentar um livro funcional e simples que atenda aos alunos que cursam a disciplina de Econometria.

Para contornar a totalidade dos conteúdos do livro, sugerimos dividi-lo em duas partes ou em duas disciplinas.

Parte I

Na introdução definimos por que a econometria é uma disciplina autônoma e desenvolvemos a sua metodologia a qual foi adotada em todas as unidades. Apresentamos um exemplo motivador para despertar a curiosidade da econometria.

No capítulo 2, apresentamos uma breve introdução aos comandos do R, as funções mais importantes direcionadas a área econométrica assim com as bibliotecas utilizadas, manipulação de números, vetores e matrizes, multiplicação e inversão de matrizes. A leitura de dados de arquivos em vários formatos, construção de gráficos como histogramas, caixas (boxplot), setores, densidades e de distribuição, funções acumuladas, gráfico de setores, plotagem de ajuste

dos modelos de regressão e testes estatísticos mais usados como: Qui-quadrado, t-student, F.

No capítulo 3 abordamos o modelo de regressão linear simples, os seus pressupostos básicos e o modelo matemático e estendemos ao modelo de regressão múltipla. Tratamos matricial os modelo de regressão, calculamos as estimativa de parâmetros e seus respectivos intervalos de confiança. São avaliadas também as perturbações ou resíduos estocásticos, além disso apresenta-se a técnica de mínimos quadrados ordinários e de máxima verosimilhança, para as estimativas dos parâmetros e suas propriedades. Definimos o teorema de Gaus - Markov, coeficientes de correlação e de determinação para indicar a força da causalidade. Simulação de modelos de regressão utilizando experimentos de Montecarlo com e sem violação de pressupostos básicos, Análise de variância (ANOVA).

No capítulo 4 aborda-se a linguagem de programação C, construindo replicas de amostras com os experimentos de Montecarlo para a valiar a normalidade dos erros no modelo de regressão linear simples utilizando diferentes erros do tipo I.

No capítulo 5 é tratado casos especias de regressão como ANOVA em delineamentos, como inteiramente causalizado e bloco causalizado, assim como modelos de efeitos fixos e aleatórios.

No capítulo 6 atenção é dada para a superação dos pressupostos básicos dos modelos de regressão como Heterocedasticidade, Multicolinearidade, Autocorrelação, tratando a natureza, técnicas de identificação e correção quando violado algum pressuposto.

Parte II

Na segunda parte, iniciando o capítulo 7, apresenta-se propriedades dos estimadores e alguns algoritmos de simulação para determinar o desempenho de estimadores como viés, consistência e EQM.

No capitulo 8 iniciamos com técnicas de otimização como Newon, Quase Newton, Steepest Ascend, Newton Rapshon, Método de Newton, Scoring de Fisher, Quase- Newton, BFGS. São apresentados os modelos não lineares entre estes os modelos exponenciais e logísticos, também aborda-se modelos de regressão para respostas binárias, em particular usamos os modelos não lineares como logit, probit, clog-log.

O capítulo 9 correspondente a séries temporais, identificamos uma série usando a decomposição dos seus elementos, como sazonalidade e tendência. Abordamos o deflacionamento de uma série e métodos de suavização, iniciando com os algoritmos de Holt e de Holt Winters, modelos autorregressivos (AR), médias moveis (MA) e integrados ARIMA(p, d, q), acrescentamos também modelos sazonais como o SARIMA, e modelos com presença de memoria longa ou curta tratados pela técnica ARFIMA. Para séries de alta volatilidade os modelos não lineares estruturais como ARCH e GARCH. Para determinar a integração de uma série são propostas raízes unitárias, assim com apresentamos simulações no ambiente R, como o desempenho de modelos AR e MA, para avaliar o ver-

dadeiro nível de significância em vários cenários de simulação.

O último capítulo é destinado a geração de números aleatórios, apresentamos a bateria de testes Diehard, útil para o processo de obtenção de dados em simulações, desenvolvido por George Marsaglia. Dois testes Diehard foram usados e propomos um novo gerador de números aleatórios, utilizando os dígitos do número π como processo gerador. Estes testes são implementados no código de programação C.

Capítulo 1

Introdução

1.1 Como funciona a Econometria

Dados econômicos, precisam de uma atenção estatística e matemática para dar fundamento a suas teorias. Atualmente pelo avanço computacional não há barreiras para serem aplicadas em programas disponíveis no mercado, e torna-se até divertida se definimos uma plataforma de trabalho. A econometria requer de uma abordagem diferenciada, pois faz uso de da teoria econômica, economia matemática, estatística econômica, estatística matemática, inferência estatística e recursos computacionais. Esta disciplina aproxima a Estatística da Economia, onde a teoria econômica formula as hipóteses; por exemplo: uma redução de preços de uma mercadoria deverá aumentar a quantidade demandada dessa mercadoria, considerando uma relação inversa entre preço e quantidade demandada, ou a despesa de consumo per capita depende da renda per capita, considerando uma relação direta, ou seja: a medida que a renda aumenta a despesa tende a aumentar. Outro exemplo onde consideremos a área microeconômica: consumo de água de uma residência e o valor pago. É natural imaginar que há uma relação direta entre consumo e valor pago pelo consumo, porem precisamos saber qual e quanto é esta força de relação, por tanto o econometrista fornece esta informação usando a inferência estatística e a estatística econômica. As estimativas da relação causal entre duas variáveis fica por conta de estatística matemática. finalmente a necessidade do recurso computacional é superado usando a plataforma R. A econometria surge como uma disciplina autônoma em virtude de aglutinar todas estas áreas do conhecimento merecendo um estudo próprio.

1.2 Tipos de dados

Os tipos de dados são:

1. Dados de corte transversal, representam observações de uma ou mais variáveis coletadas no mesmo ponto do tempo. Por exemplo, os censos

no Brasil são coletados num instante do tempo, onde variáveis de sociodemográficas são exploradas como: contagem da população, número de indivíduos em cada família, idade dos integrantes da família ou variáveis econômicas, como: renda, consumo, tipo de moradia etc., estes dados são adquiridas a cada década pelo IBGE (`www.ibge.gov.br`).

2. Dados longitudinais ou de séries temporais, são caracterizadas por observações ao longo do tempo como por exemplo o índice geral de preços distribuição interna (IGP-DI) coletadas mensalmente, ou séries diárias como a cotação da IBOVESPA. No Brasil um site que disponibiliza dados desta natureza é o IPEADATA (`www.ipeadata.gov.br`).

3. Dados de painel, é uma mistura de dados de corte transversal e longitudinais, ou ao longo do tempo estudamos o comportamento de unidades individuais, por exemplo, séries mensal do índice de preços nos municípios sergipanos, série anuais do produto interno bruto (PIB) nas principais capitais Brasileiras, ou série anual da balança comercial dos municípios do Brasil, disponíveis no IBGE e IPEADATA.

1.3 Metodologia

Os passos da metodologia econométrica tradicional ou clássica se inicia desde a formulação da hipótese até as conclusões, e em geral são:

1.3.1 Hipótese

Como referência usaremos a hipótese nula $H0$, a qual nulifica qualquer diferença, e a hipóteses alternativa Ha, aquela que contradiz, total ou parcialmente a hipótese nula. Exemplo:

1. $H0$: não existe uma relação entre consumo de água e valor pago.

2. Ha: existe uma relação entre consumo de água e valor pago.

A hipótese alternativa pode ser dividida em três possibilidades, no exemplo do consumo de água pode ser dividida em: existe alguma relação, ou a relação é negativa ou a relação é positiva. A primeira denomina-se de hipótese alternativa com teste bicaudal, o segundo de teste unicaudal á esquerda, e finalmente o último com teste unicaudal a direita. Nosso exemplo considere Ha: existe alguma relação (teste bicaudal)

1.3.2 Modelo matemático

Usaremos uma função de consumo simples

$$\hat{Y} = \beta_1 + \beta_2 X \tag{1.1}$$

A variável dependente Y: valor pago em reais pelo consumo, é estimado por $\hat{Y}$. A variável independente é X: consumo de água em m^3. β_1, ou intercepto, representa o valor pago quando não há consumo de água. Nosso interesse é determinar a tangente ou a variação pelo consumo β_2. O número de parâmetros é $p = 2$.

1.3.3 Modelo econométrico

O modelo matemático fornece a relação determinística entre valor pago em reais e consumo de água, porem as relações entre as variáveis econômicas são em geral inexatas. Assim ao longo do tempo no podemos esperar que o valor pago e o consumo em m^3 se situem exatamente numa linha reta, isto por que além do consumo, outras variáveis afetam o consumo, por exemplo o tamanho da família, a reforma da casa, ou a inflação presente, podendo alterar o valor pago pelo consumo, os efeitos destas variáveis externas que não podem ser controladas constituem o erro aleatório do modelo.

1.3.4 Nível de significância

Denominado de α, frequentemente usa-se $\alpha = 0.05$; em situações de risco, usa-se $\alpha = 0.01$. Em testes bicaudais divide-se α em duas partes, e localizam-se nos extremos da distribuição de prova. Para testes unicaudais localiza-se no extremo da distribuição indicada pela hipótese alternativa. Nosso exemplo $\alpha = 0.05$ e por tanto $\alpha/2 = 0.025$.

1.3.5 Obtenção de dados

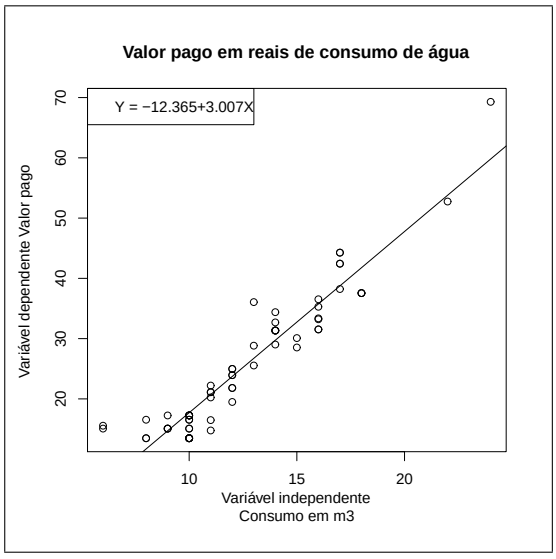


figura 1. Diagrama de dispersão.

Considere 65 observações (mês a mês) do consumo em m^3 de água e valor pago em reais em uma residência, durante o período de 12/2005 a 04/2011, ver plotagem na figura 1. Podemos observar visualmente que há uma relação positiva entre valor pago e consumo de água, onde a variação pelo consumo (tangente) β_2, cresce positivamente. Precisamos verificar se esta variação β_2 é estatisticamente diferente de zero (significativa). Ver dados no apêndice A.

1.3.6 Estatística de prova

Denominada de função pivô ou estatística de teste. No âmbito paramétrico usa-se as estatísticas: t student (amostras pequenas), quando n, é grande é indiferente o uso da t student ou normal. No exemplo para testar qual a inclinação da variação de consumo de água, usamos a seguinte estatística:

$$t_0 = \frac{\hat{\beta}_2}{dp(\hat{\beta}_2)} \approx t_{n-2gl} \tag{1.2}$$

onde: t_0 é o valor calculado da estatística t, $\hat{\beta}_2$ é a estimativa do parâmetro da variação do valor pago β_2, $dp(\hat{\beta}_2)$ é o desvio padrão da estimativa do parâmetro β_2, o valor t_0 segue uma distribuição t-student com $n-2$ graus de liberdade(gl). Os graus de liberdade é calculado diferenciando o tamanho da amostra com o número de parâmetro do modelo $(n-p)$. O valor quantílico $t_{n-2gl}(0.025)$, corresponde a distribuição t-student com 0.025 de significância com $n-2$ graus de liberdade, e n é o número total de observações. Nosso exemplo $\beta_2 = 3.007$, e $dp(\beta_2) = 0.13$, por tanto $t_0 = 23.13$ é comparado com o valor da tabela da t-student com 63 gl é com $\alpha/2 = 0.025$, sendo este valor igual 1.9983.

1.3.7 Tipos de erro

Ocorre o erro tipo I (α) ao se rejeitar a hipótese de nulidade quando deveria aceitá-la. Por outro lado quando se aceita a hipótese de nulidade quando deveria rejeitá-la comete-se o erro tipo II (β). Nas duas situações tomou-se uma decisão errada. O interesse é a minimização dos erros tipo I e II que na prática só é possível aumentando-se o tamanho da amostra. Por razões diversas o aumento do tamanho da amostra muitas vezes torna-se impraticável. A gravidade do erro tipo I é distinta da do erro tipo II, quando se consegue identificar o erro mais grave a opção é a minimização deste, porem quando a probabilidade de ocorrência de um tipo de erro diminui a do outro aumenta e vice versa. Podemos observar este critério na seguinte tabela:

	Não rejeita	Rejeita
H0 verdadeira	Correto $(1-\alpha)$	Erro do tipo I (α)
H0 falsa	Erro do tipo II (β)	Correto $(1-\beta)$

1.3.8 Regra de decisão

A plataformas R, em todas suas saídas estatísticas apresenta o valor observado da função pivô e o $p-valor$ (Área de probabilidade está relacionada

como nível de significância), assim a função pivô apresenta o valor numérico na função de densidade da estatística. Se o *pvalor* é menor do que $\alpha = 0.05$ no caso do teste unicaudal ou $\alpha = 0.025$ no caso bicaudal, concluímos que não há evidencia suficiente para aceitar H0. Caso contrario se o $p-valor$ for maior que $\alpha = 0.05$, podemos concluir que não há evidencia suficiente para rejeitar H0. Nosso exemplo: $t_0 = 23.13 > 1.998$, (1.998 é valor da distribuição t com 63 gl e nível de significância bicaudal de 0.05), este valor localiza-se na cauda positiva da distribuição com 63 graus de liberdade($n - 2 = 65 - 2$). Por tanto não há evidencias para aceitar H0, em favor de aceitar que o coeficiente de inclinação (ou tangente) é positiva. No recorrer do livro usaremos códigos para as medidas do $p - valor$, que em geral aparece em todo comando `summary()`. Como consequência a linha que aparece a seguir será excluída das saídas do R:

```
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1  1
```

será interpretada da seguinte maneira:

```
0     ***  significância menor que 0.001
0.001  **  significância menor que 0.01
0.01    *  significância menor que 0.05
0.05    .  significância menor que 0.1
0.1        significância menor que 1
```

1.3.9 Conclusões

Esta parte do teste de hipóteses é explicada em conjunto: O Economista que é o profissional onde as observações foram coletadas e o Estatístico que operacionaliza as técnicas matemáticas que fornecem a validação de uma determinada teoria, decidem sobre as providencias futuras. No exemplo podemos afirmar que a cada metro cubico a mais de consumo de água será pago em média 3 reais a mais. No momento este valor pago pode parecer alto, porem na falta de este liquido elementar, este valor pode aumentar e por tanto sofrerá mudanças.

1.3.10 Previsões

Muitas vezes estamos interessados em determinar o que acontece quando uma quantidade de consumo de água em m^3 é utilizada e qual o valor pago em reais. Por exemplo se nosso interesse é saber quanto será pago por $20m^3$ de consumo de água em uma residência. A conta a realizar é:
Valor pago $= -12.365 + 3.007 * 20 = 47.775$ reais
A interpretação deste valor é: em média o valor pago em reais por $20m^3$ de água consumida é de aproximadamente 47.8 reais

1.3.11 Resíduos

Uma questão fundamental em econometria é identificar o comportamento dos erros aleatórios que na prática é estimado pelos resíduos do modelo. Considere Y, o valor real pago e $\hat{Y}$ valor pago estimado a partir de um modelo, então

o erro é aleatório é definido por: $\epsilon_i = Y_i - \hat{Y}_i$.

A abordagem mais formal dos erros é feita nos próximos capítulos. Uma questão importante é determinar qual o comportamento da estimativa destes erros aos quais chamamos de resíduos. Para determinar a normalidade dos resíduos realizamos o teste de **Jarque Bera**, com a função `jarque.bera.test()` da biblioteca `tseries`, com o seguinte gráfico:

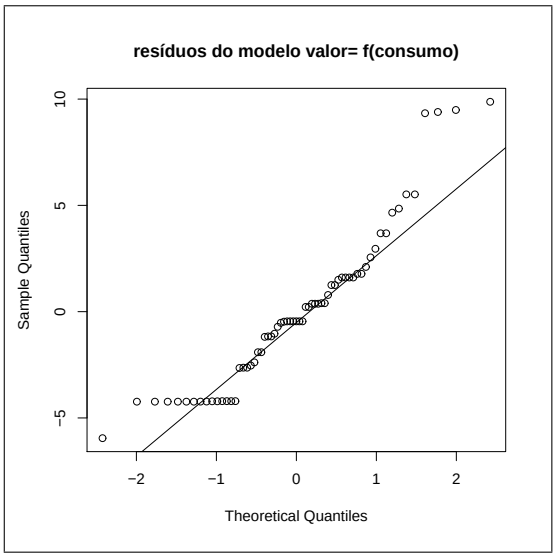


Figura 2. Normalidade dos resíduos

1.4 Exercícios

1. Por que a econometria precisa ser uma disciplina autônoma?
2. Que é econometria?
3. Pesquise dados de corte transversal, corte longitudinal e de painel.
4. Construa uma metodologia econométrica para o consumo familiar em função da renda familiar (linear).
5. No item anterior é possível encontrar uma relação quadrática?
6. Quais são os pressupostos básicos de um modelo de regressão linear simples?
7. Como é chamado o parâmetro de inclinação ou tangente num modelo de regressão linear simples em econometria?
8. Que é o projeto R?
9. R é um programa econométrico?

Capítulo 2

Alguns comandos do R

R é um sistema de estatística computacional e gráfica, desenvolvido por um grupo de referis, que podem ser visitados no site: `http://www.r-project.org`. Este projeto de software segue as orientações do Geral Public License(GPL). Para instalar o programa do R temos que estar conectado a internet e baixar o programa no site, `http://cran.r-project.org`.

2.1 Iniciando o R

para instalar pacotes do R uma lista completa esta disponível com o comando:

```
install.packages()
```

Quando instalamos o R uma lista de pacotes padrão já estão na biblioteca:

```
search()
[1] ".GlobalEnv"        "package:stats"     "package:graphics"
[4] "package:grDevices" "package:utils"     "package:datasets"
[7] "package:methods"   "Autoloads"         "package:base"
```

2.2 Dados permanentes e removendo objetos

Durante uma sessão no R, objetos são criados e guardados por nomes, para verificar estes objetos usamos o comando ou função

```
objects( ) # ou ls( )
```

```
[1] "a1"               "a2"             "aa"          "acei"
[5] "ads"             "aju"           "aju1"        "ajuste"
[9] "ajuste1"      ajuste2"       "ajustehw"           "ara"
```

Para remover objetos antigos ou sem uso utilizamos o comando rm:

```
rm(a1, a2, aa, acei, ads, aju)
```

Para detectar os pacotes na biblioteca local

```
library()
```

Para instalar o pacote `urca` disponível na biblioteca:

```
install.packages("urca")
```

Um pacote instalado tem que ser chamado no R para realizar a tarefa. Para chamar o pacote `urca` já instalado.

```
library(urca)
```

Para verificar os pacotes na biblioteca externa.

```
update.packages()
```

Para ver a lista de pacotes instalados na biblioteca local:

```
installed.packages()
```

2.3 Manipulação de números e vetores

Para declarar números podemos usar o seguinte comando

```
v=10
v
[1] 10
```

Alertamos que no R o símbolo $<-$ é equivalente ao símbolo $=$.
Para criar um vetor com os números 1, 3, 3.4, 6, 99, 3, com o nome q, usamos:

```
q=c(1,3,3.4,6,99,3)
```

Outra forma alternativa é usando o comando:

```
assign("q",c(1,3,3.4,6,99,3))
```

Comandos de soma, diferença, multiplicação e divisão são usados com os sinais de +, -, *, /, para inverter q, podemos usar:

```
1/q
[1] 1.0000000 0.3333333 0.29411765 0.1666667 0.0101010 0.3333333
```

Para determinar como verdadeiro ou falso para os valores de q maiores do que 3 podemos realizar:

```
q>3
[1] FALSE FALSE TRUE TRUE TRUE FALSE
```

para restringir valores de q maiores do que 3:

```
q[q>3]
[1] 3.4 6 99
```

Podemos converter números em caracteres ou dígitos assim:

```
z2=c(0:9) # equivalente a z2=0:9
digit=as.character(z2)
digit
 [1] "0" "1" "2" "3" "4" "5" "6" "7" "8" "9"
```

Para retornar dígitos em números com o seguinte comando:

```
d=as.integer(digit)
d
 [1] 0 1 2 3 4 5 6 7 8 9
```

Para preparar uma variável para ingressar seus valores por exemplo

```
e=numeric()
e[3]=17
```

Onde o terceiro elemento de e é 17 e os dois primeiros ainda não definidos:

```
e
[1] NA NA 17
```

É possível a escolha de elementos de 2 em 2 ou de 3 em 3 a seguir:

```
alpha=z[2*1:5]
alpha
[1]  2  4  0 NA NA
z
[1] 1 2 3 4 5 0
alpha=z[3*1:5]
alpha
[1]  3  0 NA NA NA
```

Construímos um vetor qualitativo com 10 homens e 5 mulheres asi:

```
w=c(rep("homem",10), rep("mulher",5))
w
r=c(rep(50,10), rep(30,4))
r
```

Considere o tamanho do calçado no pé de cada individuo em w:

```
pé=c(40,42,39,41,44,40,42,41,44,40,35,35,37,36,37)
```

Podemos identificar a distribuição de frequências de w e pé da seguinte forma:

```
table(z,pé)
        pé
z        35 36 37 39 40 41 42 44
  homem   0  0  0  1  3  2  2  2
  mulher  2  1  2  0  0  0  0  0
```

Para construir gráficos para as tabelas usamos as funções:

```
plot(table(z,pé))
hist(table(z,pé))
barplot(table(z,pé))
```

2.4 Funções matemáticas

Para utilizar as funções matemáticas, usamos os comandos:

```
history()   # comandos usados previamente
length(x)   # número de elemetos do vetor ou lista x
log(x)      # log na base e de x
exp(x)      # antilogaritmo de x
log(x,n)    # log na base n de x
log10(x)    # log na base 10 de x
sqrt(x)     # raiz quadrada de x
factorial(x)# factorial de x(x!)
choose(n,x) # combinatoria:  n!/(x!(n-x)!)
gamma(x)    # x (x-1)!, x inteiro
lgamma(x)   # log natural da gamma em x
floor(x)    # menor inteiro de x
trunc(x)    # escolha do menor inteiro
abs(x)      # valor absoluto de x
cos(x)      # cosseno de x
sin(x)      # seno de x
tan(x)      # tangente de x
acos(x) # inversa cosseno
asin(x) # inversa seno
atan(x)     # inversa tangente
acosh(x)    # inversa cosseno hiperbólica
asinh(x)    # inversa seno hiperbólica
atanh(x)    #inversa tangente hiperbólica
round(x, digits=2)  # aproximação a duas casas decimais
signif(x, digits=6) # x com notação científica de 6 dígitos
```

2.5 Funções estatísticas

```
max(x)     # valor máximo de x
min(x)     # valor mínimo de x
```

```
sum(x)     # soma total dos valores de x
mean(x)    # média aritmética de x
median(x)  # mediana de x
range(x)   # vetor de mínimo e máximo de x
var(x)     # variância de x
cor(x,y)   # correlação de x e y
sort(x)    # ordena em forma crescente os elementos de x
rank(x)    # vector de filas em x
order(x)   # valores originais de x
quantile(x)# vetor contendo os quartis de x
cumsum(x)  # vetor contendo a suma total elemento a elemento
cumprod(x) # vetor contendo o produto elemento a elemento
cummax(x)  # vetor não decrescente de x, substituindo valores
cummin(x)  # vetor não crescente de x, substitui os valores
             pelo mínimo
pmax(x,y,z)# vetor contendo o vetor maior entre x y ou z, e
             substituindo o máximo em cada posição.
pmin(x,y,z)# vetor contendo o vetor maior entre x y ou z,
             e substituindo o mínimo em cada posição.
colMeans(x)# calcula a média por colunas na matriz x
colSums(x) # calcula os totais por coluna na matriz x
rowMeans(x)# calcula a média por linhas na matriz x
rowSums(x) # calcula os totais por linhas na matriz x
```

2.6 Cores

O R disponibiliza cores usando a função `palette()` com 8 cores, outra função é `colours()`, e que também podem ser identificados por um número já especificado por exemplo:

```
t=(1:20)*0.92
plot(t,sin(t), col="red", pch=16,cex=2) #ou
plot(t,sin(t), col=2, pch=16,cex=2)
```

Onde `cex`, representa o carácter de expansão da bolinha (quanto maior o número maior a bolinha). Para incrementar várias cores no mesmo gráfico podemos realizar o seguinte comando:

```
plot(t,sin(t), col=1:20, pch=16,cex=2)
```

Para modificar a bolinha apenas modificamos pch:

```
plot(t,sin(t), col=1:20, pch="+",cex=2)
```

Cores podem também ser incrementadas fora do corpo do gráfico

```
plot(t,sin(t), main="seno", col.main=4, col=1:20, pch="+",cex=2)
```

Outras alterações de cores podem ser construídas com:

```
col.main    # cor para o título do gráfico
col.axis    # cor para os números dos eixos
col.lab     # cor para os títulos dos eixos
col.sub     # cor para o subtítulo do gráfico
```

A fonte pode ser alterada da seguinte maneira:

```
font.main    # fonte para o título do gráfico
font.axis    # fonte para os números dos eixos
font.lab     # fonte para os títulos dos eixos
font.sub     # fonte para o subtítulo do gráfico
```

Onde: 1 é texto normal, 2 preto, 3 itálico, 4 preto itálico para simbolo fonte. A orientação dos números pode ser alterada com o comando *las*=; onde: 1 horizontal a x, 2 perpendicular a x, 3 perpendicular a ambos eixos).

2.7 Fatores ordenados e desordenados

Exemplo

```
estados=c("PE","AL","SE","BA","PB","BA","PB","AL","SE","BA","PB",
+ "SE", "BA","SE","BA")
estadosf=factor(estados)
estadosf
 [1] PE AL SE BA PB BA PB AL SE BA PB SE BA SE BA
Levels: AL BA PB PE SE
levels(estadosf)
[1] "AL" "BA" "PB" "PE" "SE"
salarios=c(60,49,59,56,49,48,56,59,60,69,66,58,47,58,68)
```

A função tapply é usada para reordenar vetores, calculando uma característica dos níveis, no exemplo suponha que os salários médio e o desvio padrão por cada trabalhador por dia nos estados seja dado a seguir:

```
mestados=tapply(salarios,estadosf,mean)  #média
mestados
   AL    BA    PB    PE    SE
54.00 57.60 57.00 60.00 58.75
destados=tapply(salarios,estadosf,sd)   #desvio padrão
destados
        AL         BA         PB         PE         SE
 7.0710678 10.5498815  8.5440037         NA  0.9574271
```

Construímos a função do coeficiente de variação de duas formas distintas, e incluímos no comando tapply:

```
cv=function(x) sd(x)*100/mean(x)
cvestados=tapply(salarios,estadosf,cv)
cvestados
       AL        BA        PB        PE        SE
13.094570 18.315767 14.989480        NA  1.629663

cv=function(x) sqrt(var(x))*100/mean(x)
cvestados=tapply(salarios,estadosf,cv)
cvestados
       AL        BA        PB        PE        SE
13.094570 18.315767 14.989480        NA  1.629663
```

2.8 Vetores e Matrizes

Um vetor pode ser considerado como a entrada de uma coleção de dados, sendo estes numéricos ou não. R facilita a manipulação e construção de vetores, em particular casos como matrizes. Considere a construção de 30 números aleatórios Uniformes,

```
z=runif(30)
dim(z)=c(3,5,2)
z
, , 1

          [,1]      [,2]       [,3]       [,4]      [,5]
[1,] 0.9500783 0.9262226 0.05847740 0.34257312 0.3718370
[2,] 0.8591887 0.7057183 0.20592054 0.03569694 0.6709746
[3,] 0.6806468 0.6270572 0.01570158 0.21650180 0.5170189

, , 2

          [,1]      [,2]       [,3]      [,4]      [,5]
[1,] 0.3539723 0.5918548 0.37104534 0.2148440 0.4169542
[2,] 0.8895178 0.5751237 0.69399721 0.3808024 0.3872492
[3,] 0.8036340 0.7065130 0.03400331 0.1593949 0.2306890
```

2.8.1 Indexando matrizes

```
x=array(1:20,dim=c(4,5))
x
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9   13   17
[2,]    2    6   10   14   18
[3,]    3    7   11   15   19
[4,]    4    8   12   16   20
i=array(c(1:3,3:1),dim=c(3,2))
```

```
i
     [,1] [,2]
[1,]    1    3
[2,]    2    2
[3,]    3    1
x[i]
[1] 9 6 3

# x[1,3]=9; x[2,2]=6; x[3,1]=3
```

O índice i indica os valores da matriz x a serem extraídos, temos

```
x[i]=0  # Substituímos x[i] por zero.
x
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    0   13   17
[2,]    2    0   10   14   18
[3,]    0    7   11   15   19
[4,]    4    8   12   16   20
```

Construindo uma matriz 3×3 com elementos de 1 ao 9.

```
c=matrix(c(1:9),3,3)
c
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
```

2.8.2 Sequências

Sequencias são construídas, ingressando o primeiro elemento, o ultimo elemento, e o espaçamento.

```
b=seq(10,14, by=.5)
b
[1] 10.0 10.5 11.0 11.5 12.0 12.5 13.0 13.5 14.0
a=seq(1,9)
ab=a%*%b # multiplicando vetores
ab
     [,1]
[1,]  570
ab=outer(a,b,"*")
ab
      [,1] [,2] [,3]  [,4] [,5]  [,6] [,7]  [,8] [,9]
 [1,]   10 10.5   11  11.5   12  12.5   13  13.5   14
 [2,]   20 21.0   22  23.0   24  25.0   26  27.0   28
 [3,]   30 31.5   33  34.5   36  37.5   39  40.5   42
```

```
 [4,]   40 42.0   44  46.0   48  50.0   52  54.0   56
 [5,]   50 52.5   55  57.5   60  62.5   65  67.5   70
 [6,]   60 63.0   66  69.0   72  75.0   78  81.0   84
 [7,]   70 73.5   77  80.5   84  87.5   91  94.5   98
 [8,]   80 84.0   88  92.0   96 100.0  104 108.0  112
 [9,]   90 94.5   99 103.5  108 112.5  117 121.5  126
```

```
ab=a%*%t(b)
ab
      [,1] [,2] [,3]  [,4] [,5]  [,6] [,7]  [,8] [,9]
 [1,]   10 10.5   11  11.5   12  12.5   13  13.5   14
 [2,]   20 21.0   22  23.0   24  25.0   26  27.0   28
 [3,]   30 31.5   33  34.5   36  37.5   39  40.5   42
 [4,]   40 42.0   44  46.0   48  50.0   52  54.0   56
 [5,]   50 52.5   55  57.5   60  62.5   65  67.5   70
 [6,]   60 63.0   66  69.0   72  75.0   78  81.0   84
 [7,]   70 73.5   77  80.5   84  87.5   91  94.5   98
 [8,]   80 84.0   88  92.0   96 100.0  104 108.0  112
 [9,]   90 94.5   99 103.5  108 112.5  117 121.5  126
```

```
d=outer(0:9,0:9)
d
        [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
 [1,]    0    0    0    0    0    0    0    0    0     0
 [2,]    0    1    2    3    4    5    6    7    8     9
 [3,]    0    2    4    6    8   10   12   14   16    18
 [4,]    0    3    6    9   12   15   18   21   24    27
 [5,]    0    4    8   12   16   20   24   28   32    36
 [6,]    0    5   10   15   20   25   30   35   40    45
 [7,]    0    6   12   18   24   30   36   42   48    54
 [8,]    0    7   14   21   28   35   42   49   56    63
 [9,]    0    8   16   24   32   40   48   56   64    72
[10,]    0    9   18   27   36   45   54   63   72    81
```

2.8.3 Multiplicação e inversão de matrizes

```
a= matrix(c(1:9),3,3)
a
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
b= aperm(A,c(2,1))
b
     [,1] [,2] [,3]
```