Todo list

Sérgio da Silva Rodrigues

Aplicação de Aprendizagem por Máquinas para estimação de preço de imóveis na cidade do Rio de Janeiro

Rio de Janeiro
2015

Sérgio da Silva Rodrigues

Aplicação de Aprendizagem por Máquinas para estimação de preço de imóveis na cidade do Rio de Janeiro

Dissertação apresentada a Fundação Getúlio Vargas, para a obtenção de Título de Mestre em Modelagem Matemática Aplicada da Informação, na Escola de Matemática Aplicada.

Orientador: Renato Rocha Souza

Rio de Janeiro, RJ
2015

Sérgio da Silva Rodrigues
Aplicação de Aprendizagem por Máquinas para estimação de preço de imóveis na cidade do Rio de Janeiro
37 páginas
Dissertação (Mestrado) - Escola de Matemática Aplicada - Fundação Getúlio Vargas.

1. Aprendizagem por Máquinas
2. Regressão Linear
3. Modelo Hedônico

I. Fundação Getúlio Vargas. Escola de Matemática Aplicada.

Comissão Julgadora:

Prof. Dr.
Nome

Prof. Dr.
Nome

Prof. Dr.
Nome do Orientador

Dedicatória...

Exemplo de epígrafe

O que é bonito?
É o que persegue o infinito;
Mas eu não sou
Eu não sou, não...
Eu gosto é do inacabado,
O imperfeito, o estragado, o que dançou
O que dançou...
Eu quero mais erosão
Menos granito.
Namorar o zero e o não,
Escrever tudo o que desprezo
E desprezar tudo o que acredito.
Eu não quero a gravação, não,
Eu quero o grito.
Que a gente vai, a gente vai
E fica a obra,
Mas eu persigo o que falta
Não o que sobra.
Eu quero tudo que dá e passa.
Quero tudo que se despe,
Se despede, e despedaça.
O que é bonito...

Lenine e Bráulio Tavares

Agradecimentos

Agradeço ao meu orientador, ao meu co-orientador, aos meus colaboradores, aos técnicos, à seção administrativa, à fundação que liberou verba para minhas pesquisas, aos meus amigos, à minha família e ao meu grande amor.

Resumo

Esta, quem sabe, é a parte mais importante do seu trabalho. É o que a maioria das pessoas vai ler (além do título). Seja objetivo sem perder conteúdo. Um bom resumo explica porquê este trabalho é interessante, relata como foi feito, o que foi encontrado, contextualiza os resultados e delineia conclusões.

Palavras-chave: palavra1, palavra2, palavra3

Abstract

This is the most important part of your work. This is what most people will read. Be concise without omitting content. A good abstract explains why this is an interesting study, tells how it was done, what was found, contextualizes the results and set conclusions.

Keywords: word1, word2, word3

Lista de Figuras

Lista de Tabelas

List of Listings

Sumário

Capítulo 1

Introdução

Desde o anúncio da realização de dois dos principais eventos esportivos da era moderna na cidade do Rio de Janeiro, a Copa do Mundo em 2014 e as Olimpíadas de Verão em 2016, os preços dos imóveis residenciais e comerciais tiveram uma alta histórica. Por exemplo, o bairro do Leblon figura como o metro quadrado de mais alto valor do Brasil , enquanto que outros bairros suburbanos também apresentam valores acima da média nacional. Em decorrência do aquecimento do mercado, vários empreendimentos imobiliários surgiram ao longo da cidade, tendo como a Barra da Tijuca e Jacarepaguá os bairros de maior concentração destes, em função da falta de espaço em locais mais tradicionais como aqueles na Zona Sul e Tijuca.

colocar referencia

Entretanto, muitos destes empreendimentos apresentam preços considerados elevados, o que nos leva a perguntar como estimar o valor de um novo imóvel novo ou usado em um determinado local. No passado, pesquisas relacionadas a esse tema utilizavam uma ferramenta estatística conhecida como Modelo Hedônico para a estimação de preço de um bem com base em suas partes constituintes. Em se tratando de imóveis, considerava-se suas partes constituintes como número de dormitórios, banheiros, suítes, existência de varanda, área construída, posição do apartamento em relação à rua, andar, vagas de garagem, entre outros. As observações eram então submetidas a um estimador,

que na Matemática Aplicada é conhecido como Regressão Linear, e o resultado é um conjunto de coeficientes a serem aplicados em novas observações para a estimação do preço. Uma das limitações deste método é ausência da localização espacial das observações e, consequentemente, a carência de apreciação da autocorrelação espacial dos preços.

Nessa dissertação, lançamos mãos de técnicas de mineração de dados para uma análise exploratória dos preços dos imóveis na cidade do Rio de Janeiro, onde avaliaremos a assertividade da regressão linear com e sem localização espacial, e finalmente consideraremos os efeitos da autocorrelação espacial. .

colocar ref

Capítulo 2

Revisão da Literatura

Neste capítulo apresentamoas a revisão bibliográfica das teorias utilizadas neste estudo.

2.1 Regressão Linear

Fenômenos da natureza, ou aqueles provocados pela ação do homem, podem ser estudados decompondo-os em variáveis numéricas ou categóricas, a fim de observar as relações entre elas e, possivelmente, identificar padrões comportamentais ou estimar resultados com base em suposições. Essas relações podem ser determinadas elencando-se um ou mais dessas variáveis como as variáveis de interesse a serem expressas em função das demais variáveis restantes. Ao longo desse estudo denominaremos as variáveis de interesse por **variáveis dependentes**, cujo nome apropriadamente indica uma relação de dependência com as demais variáveis, denominadas **variáveis independentes**, (Andersen e Skovgaard, 2010, p.2). Como o escopo desse estudo limita-se a apenas uma variável independente, representamos uma observação qualquer dessa variável por y e as respectivas n variáveis independentes pelo vetor $x = (x_1, x_2, ..., x_n)$. As matrizes $Y_{m,1}$, $X_{m,n}$ representam o conjunto de m observações das variáveis dependente e independentes, respectivamente, sendo uma determinada observação i indicada por

y_i, x_i, $i \in \{1,2,...,m\}$ e uma determinada variável independente j por x_j , $j \in \{1,2,...,n\}$. Finalmente, x_{ij} representa uma observação específica i da variável independente j.

Um dos objetivos da compreensão de um fenômeno é a capacidade de estimar um valor da variável dependente, indicado por $\hat{y}$, a partir de uma nova observação $x \notin X$, esperando-se seguir as relações naturalmente presentes em Y e X. Quando $\hat{y}$ pode assumir um valor contínuo, $\hat{y} \in \mathbb{R}$, chamamos a essa estimação de **Regressão** (Bishop, 2006, p.3), (Hastie et al., 2013, p.4). A estimação de variáveis dependentes categóricas, aquelas que representam a pertinência a um determinado conjunto, é chamado **Regressão Logística** e não é escopo desse estudo.

(Hastie et al., 2013, p.44),(Bishop, 2006, p.138) e (Murphy, 2012, p.127) definem **Regressão Linear** a classe de modelos cuja função de regressão da variável dependente $\hat{y}$ é uma combinação linear dos parâmetros $\beta_i \in \mathbb{R}$:

$$\hat{y} = \beta_0 + \beta_1 x_1 + ... + \beta_n x_n \tag{2.1}$$

Segundo (Andersen e Skovgaard, 2010, 9-11), para os casos em que x_j é categórica, $x_j \in c_1,c_2,...,c_k$, não a utilizamos diretamente no modelo, mas a substituímos por variáveis **dummy**[1], assim denominadas para representar a pertinência à categoria identificada em x_j. Tal substituição é feita criando-se k variáveis indicadoras para as $k+1$ categorias possíveis de x_j:

$$I_{1,...,k}(x_j) = \begin{cases} 1, \text{se } x_j = c_r \\ 0, \text{senão} \end{cases} \tag{2.2}$$

O coeficiente β_0 na equação 2.1 representa um deslocamento fixo do modelo, valor a ser assumido para o caso em que $\forall j : x_j = 0$, denominado *bias*[2]. Por conveniência,

[1] Sem tradução direta para a Língua Portuguesa.

[2] Não há tradução clara do significado desse termo conforme (Bishop, 2006, p.138) para a Língua Portuguesa.

assumimos uma nova variável independente $x_0 = 1$, fazendo o conjunto de variáveis independentes ter dimensões $X_{m,n+1}$, com o propósito de reduzir a equação 2.1 para a forma:

$$\hat{y} = \sum_{j=0}^{n} \beta_j x_j \tag{2.3}$$

Entretanto os parâmetros β são desconhecidos e também precisam ser estimados. Podemos fazê-lo a partir de um subconjunto das observações Y e X, a quem denominamos **conjunto de treinamento** (Bishop, 2006, p.4), (Hastie et al., 2013, p.1). A utilização dos valores atuais Y de forma a permitir uma avaliação da eficiência da estimação de β classifica esse tipo de aprendizado como **supervisionado** (Hastie et al., 2013, p.2). Reciprocamente, aprendizados **não supervisionados** são aqueles que procuram identificar estruturas em X e não dependem de Y para avaliar o aprendizado.

Segundo (Hastie et al., 2013, p.12), um dos métodos mais populares utilizado para a estimação de β, conhecido como **Método dos Mínimos Quadrados**[3], consiste em minimizar a soma dos quadrados dos erros residuais $\epsilon_i = y_i - \hat{y}_i$:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{m} (\sum_{j=0}^{n} \beta_j x_{ij} - y_i)^2 \tag{2.4}$$

Importante notar que o erro residual não é a distância euclidiana entre y e $\hat{y}$, mas tão somente a diferença escalar entre as duas variáveis, como pode ser visto na fig. 2.1.

A equação 2.4 pode ser descrita em forma matricial, onde denominamos $RSS(\beta)$ a *Soma dos Resíduos Quadrados*[4]:

[3] Do inglês *Least Squares*, tradução nossa.
[4] Do inglês *Residual Sum of Squares, RSS* (Hastie et al., 2013, p.12)

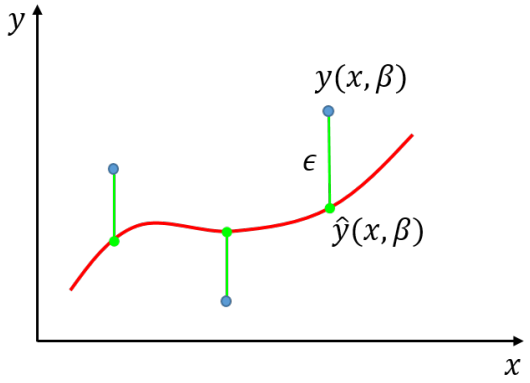


Figura 2.1: Intuição do método Mínimos Quadrados. Fonte: (Bishop, 2006, p.6), adaptado.

$$\begin{aligned} RSS(\beta) &= \sum_{i=1}^{m}(\sum_{j=0}^{n}\beta_j x_{ij} - y_i)^2 \\ &= \sum_{i=1}^{m}(\beta^T x_i - y_i)^2 \\ &= (Y - X\beta)^T(Y - X\beta) \end{aligned} \tag{2.5}$$

Derivando-se (2.5) com com respeito a β temos:

$$0 = X^T(y - X\beta) \tag{2.6}$$

Se $X^T X$ for não singular, então a solução única é dada por:

$$\hat{\beta} = (X^T X)^{-1} X^T Y \tag{2.7}$$

Se $X^T X$ for singular, então a equação 2.7 admite mais de uma solução, o que indica interdependência linear entre as variáveis X. Deparamo-nos então com uma das primeiras premissas para a utilização do Método de Mínimos Quadrados para estimação

de β, que é a independência linear entre as variáveis independentes.

Finalmente, de posse de uma nova observação $z \notin X$ podemos estimar o valor da variável dependente $\hat{y}(z)$ com:

$$\hat{y}(z) = z^T \beta \tag{2.8}$$

(Bishop, 2006, p.140-143) e (Andersen e Skovgaard, 2010, p.178-180) demonstram que o Método dos Mínimos Quadrados é derivado da Estimativa por Máxima Verossimilhança sob a premissa de que o erro residual $e_i = y_i - \hat{y}_i$ segue uma distribuição Normal.

Sob determinadas condições das escolhas das variáveis independentes em X e o número m de observações, podemos incorrer em um problema denominado **Sobreajuste**[5](Bishop, 2006, p.).que implica na estimação de $\hat{\beta}$ fazer $\hat{y}(x)$ demasiadamente bem de $y(x)$ para $x \in X$ mas não aproximar bem em novas observações $x \notin X$. (Bishop, 2006, p.4-9) ilustra esse problema com uma aplicação bem simples de Regressão Linear que é ajustar uma curva polinomial de ordem M, $\hat{y} = \sum_{i=0}^{M} \hat{\beta} x^i$, a partir de dados gerados pela função $y = \sin(x)$, com um ruído aleatório aplicado. Nesse exemplo em que temos apenas uma variável em X, propomos a construção de novas variáveis $x^2, x^3, ..., x^M$. Importante notar que essas novas variáveis não apresentam dependência linear com x, respeitando a premissa para que $X^T X$ seja não singular.

Renato:Devo explicar matematicamente a causa de overfitting que envolve demonstração estatística bayesiana???

Vê-se que na fig. 2.2 que conforme M aumenta, o polinômio resultante aproxima-se a x até que para $M = 9$ o polinômio passa exatamente exatamente sobre cada um dos dados originais mas distanciará de novas observações.

De fato, a definição de sobreajuste apresentada por (Mitchell, 1997, p.67, adaptado) diz:

> Given a hypothesis space H, a hypothesis $\hat{y} \in H$ is said to overfit the

[5]Do inglês *Overfitting*, tradução nossa.

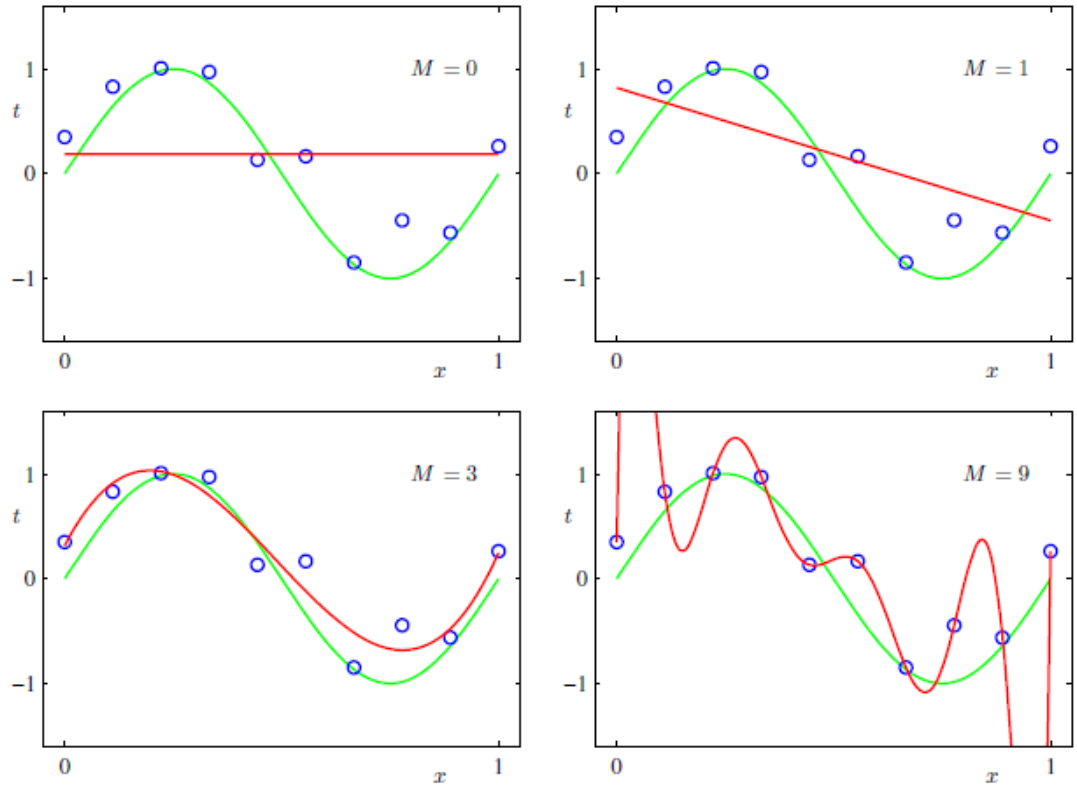


Figura 2.2: Exemplo de sobreajuste. Fonte: (Bishop, 2006, p.7), adaptado.

> training data if there exists some alternative hypothesis $\hat{y}' \in H$, such that $\hat{y}$ has smaller error than $\hat{y}'$ over the training examples, but $\hat{y}1$ has a smaller error than $\hat{y}$ over the entire distribution of instances.

Tal definição é empiricamente demonstrada na fig. 2.2 em que notamos que para $M = 3$ a soma dos erros residuais para novas observações será menor do que para $M = 9$.

Podemos verificar o sobreajuste de uma regressão medindo gráfica e numericamente o comportamento de uma medida de performance da capacidade de generalização de $\hat{y}$. Usualmente utiliza-se a *Raiz do Erro Médio Quadrático*[6] para esse objetivo, que é uma

[6]Do inglês *Root Mean Square Error*, *RMSE*, tradução nossa. Manteremos a sigla em inglês *RMSE* por conveniência.

extensão da equação 2.4 definida como (Bishop, 2006, p.7, adaptado):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{m}(\hat{y} - y)^2}{n}} \tag{2.9}$$

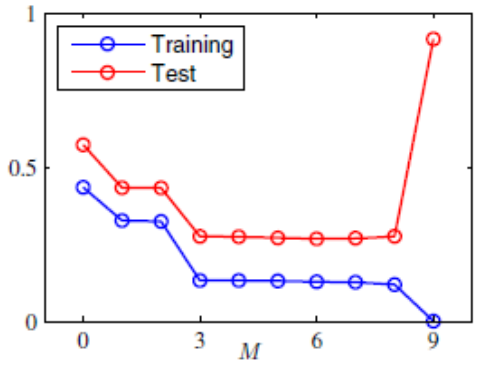
A divisão por n nos permite comparar diferentes tamanhos de conjuntos de treinamento e a raiz quadrada por sua vez garante que a $RMSE$ seja medida na mesma escala e unidade da variável dependente y (Bishop, 2006, p.7).

Entretanto, de nada adianta escolher os parâmetros cuja $RMSE$ seja mínima utilizando o próprio conjunto de treinamento para avaliação da performance pois desta forma estaremos justamente forçando o sobreajuste. Essa situação é resolvida pela técnica da **Validação Cruzada** separando os conjuntos de observações Y e X em dois subconjuntos distintos, um para treinamento, a ser denotado por Y_t e X_t, e outro para validação, denotados como Y_v e X_v (Bishop, 2006, p.32).

A fig. 2.3 demonstra a importância da Validação Cruzada na avaliação da performance da generalização de $\hat{y}$ para novas observações. Continuando com o exemplo do ajuste de um polinômio de ordem M, calculamos a raiz do erro médio quadrático, $RMSE$, para cada M, sobre o próprio conjunto de treinamento e sobre um conjunto reservado de verificação. À medida que M aumenta a performance melhora em ambos, mas para $M = 9$ fica claro o sobreajuste quando avaliado sobre o próprio conjunto de treinamento e seu impacto na performance sobre o conjunto de verificação. Um efeito prático do sobreajuste sobre os coeficientes β é esses assumirem valores abusolutos expressivos, como pode ser visto no lado esquerdo da fig. 2.3.

Em função da expressiva magnitude que os coeficientes $\hat{\beta}$ podem alcançar devido ao sobreajuste, uma alternativa é aplicar sobre a estimação de β uma penalidade proporcional ao crescimento absoluto dos próprios coeficientes, conhecido como **Regularização**[7] (Bishop, 2006, p.10,144-147) ou **Métodos de Encolhimento**[8] (Hastie et al., 2013,

[7] Do inglês *Regularization*, tradução nossa.
[8] Do inglês *Shrinkage Methods*, tradução nossa.



	$M=0$	$M=1$	$M=6$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

Figura 2.3: Intuição do soreajuste (overfitting) para a estimação de uma função polinomial em x. Fonte:(Bishop, 2006, p.8), adaptado.

p.61-69). Abaixo apresentamos a estimação de β com uma penalização denominada *Ridge* (Hastie et al., 2013, p.63):

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \left(\sum_{i=1}^{m} (y_i - \beta_0 - \sum_{j=1}^{n} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{n} \beta_j^2 \right) \tag{2.10}$$

O parâmetro λ controla o grau de penalidade a ser aplicado estimação de β. Quanto maior λ, maior será a penalidade aplicada, limitando proporcionalmente o crescimento absoluto de β. Uma penalidade excessiva pode inverter totalmente o objetivo da regularização e causar o chamado *subajuste*[9], cuja consequência também é a perda de generalização de $\hat{y}$ para novas observações, inclusive para o próprio conjunto de treinamento (Andersen e Skovgaard, 2010, p.38).

2.2 Modelos Hedônicos

Segundo Lancaster e Rosen, Modelos Hedônicos é um método de estimação dos valores implícitos das partes constituintes de um bem Long et al. (2007). Macedo (1999) cita como uma das primeiras aplicações dessa técnica a análise de preços de automóveis feita por Griliches e Dhrymes na década de 1960, decompondo automóveis em tamanho,

[9]Do inglês *Underfitting*, tradução nossa.

potência e acessórios, e também uma aplicação no mercado de imóveis na mesma década feita por Bailey, Muth e Nourse.

Ao escopo desse trabalho, Long et al. (2007) cita que Modelos Hedônicos tem sido indispensável para a avaliação do mercado imobiliário cuja importância desse tema respalda-se no impacto dos estudo macroeconômicos , no interesse de agentes governamentais como *termômetro* de fenômenos sociais como crime, trânsito, oportunidades de emprego e constituição demográfica Ismail e MacGregor (2005), além de avaliação de investimentos em benfeitorias públicas e programas sociais Long et al. (2007). De igual forma o setor privado aborda Modelos Hedônicos em precificação de imóveis para o estudo de viabilidade de empreendimentos e determinação dos itens mais valorizadas pelos consumidores Neto (2002). Os imóveis são decompostos em suas características que podem ser classificadas em três tipos (Long et al., 2007, p.3):

1. Estruturais: aquelas pertencentes unicamente ao imóvel como ano de construção, número e tipo de cômodos, posição relativa à rua de acesso, entre outras;

2. Sócio-ambientais: características derivadas por proprietários e vizinhaça como renda, desenvolvimento educacional, participação política, criminalidade;

3. Acessibilidade: demais características deriviadas diretamente da localização do imóvel como benfeitorias públicas, acesso à meios de transporte, unidades de saúde, escolas.

A relação entre o preço do imóvel e suas características é comumente avaliado por Regressão Linear, onde o valor do bem é a variável dependente e suas partes constituintes são as variáveis independentes, e o erro residual entre o valor predito e o valor real justificado em parte pelas características existentes que afetam o valor mas não expressas diretamente no modelo (Long et al., 2007, p.4). Entretanto, era discutível a observância das premissas para aplicabilidade de Regressão Linear em Modelos Hedônicos como

independência das características modeladas e do erro residual. Até a popularização dos Sistemas de Informações Geográficas, SIG[10], os efeitos derivados da localização geográfica, tal como autocorrelação espacial, não recebiam a devida atenção (Ismail, 2006, p.1). Nota-se que as características de acessibilidade e sócio ambientais podem ser compartilhadas em maior ou menor grau entre imóveis até uma certa distância (Ismail, 2006, p.3).

[10]Do inglês Geofraphic Information Systems, GIS. Traduação nossa.

Capítulo 3

Metodologia

3.1 Descrição dos dados

Variável	Definição da variável
Área	Área do imóvel
Condomínio	Valor do condomínio
Quartos	Quantidade de quartos
Suíte	Presença de suíte. Com suíte = 1, sem suíte = 0.

3.2 Obtenção dos dados

3.2.1 Informações sobre os imóveis

O ZAP Imóveis[1] é um serviço de classificados de imóveis na Internet onde proprietários de imóveis, sejam pessoas físicas ou jurídicas, anunciam-os detalhando informações que consideram relevantes a quem procura comprá-los. Essas informações foram coletadas para esse estudo pela técnica conhecida como *Web Scrapping*.

A técnica de *Web Scrapping* consiste em obter informações disponíveis na Internet, por meio de um algoritmo construído para percorrer a estrutura do recurso onde a

[1]ZAP Imóveis: http://www.zapimoveis.com.br/.

informação está, identificá-la e descobrir outros recursos que possam contê-la. Para que se possa implementar o algoritmo com as decisões adequadas à estrutura do meio é feito um estudo *a priori* buscando-se encontrar padrões de repetição e palavras chaves que se associem de alguma forma à informação desejada. Tal atividade é bastante facilitada quando o meio em questão é uma página HTML[2] que por definição[3] é um documento de texto estruturado em segmentos idenitificados por palavras chaves[4].

Dentro da linguagem de programação Python utilizada nesse estudo, recorremos à biblioteca Requests[5] para o *download* das páginas HTML do ZAP Imóveis em memória, e a biblioteca BeautifulSoap[6] para percorrer a estrutura HTML, identificar as informações desejadas e descobrir novas páginas com mais anúncios.

Obtivemos as seguintes informações para todos os imóveis:

1. Nome da rua;
2. Bairro;
3. Área cosntruída;
4. Quantidade de quartos;
5. Quantidade de suítes;
6. Quantidade de vagas de garagem;
7. Data da publicação do anúncio;
8. Valor anunciado do imóvel;

Alguns anúncios continham informações adicionais como:

[2]Do inglês *HTML: HyperText Markup Language*.
[3]Definição de HTML: http://www.w3.org/html.
[4]Do inglês *tags*. Tradução nossa.
[5]Requests: http://docs.python-requests.org/en/latest/.
[6]BeautifulSoap: http://www.crummy.com/software/BeautifulSoup.

1. Valor do condomínio;
2. Localização geográfica: latitude e longitude;
3. Características estruturais do imóvel: varanda, dependência de empregada, sala de jantar, etc;
4. Características do condomínio: piscina, sauna, portaria 24 horas, etc.

3.2.2 Distâncias dos imóveis a pontos de interesse

A fim de tentar minimizar os efeitos da autocorrelação espacial da variável preço buscamos criar variáveis de distâncias dos imóveis aos seguintes pontos de interesse:

1. Praias;
2. Lagoas;
3. Estabelecimentos de Saúde;
4. Estações de Metrô;
5. Delegacias da Polícia Civil;
6. Unidades do Corpo de Bombeiros;
7. Favelas;
8. Principais logradouros;
9. Centro da cidade do Rio de Janeiro.

A seguir detalharemos a coleta de cada uma dessas informações.

Informações obtidas do Mapa Digital do Rio de Janeiro

A localização de favelas, delegacias de Polícia Civil, unidades do Corpo de Bombeiros, principais logradouros, estações de trem e metrô foram obtidas da ferramenta Mapa Digital do Rio de Janeiro[7], mantido pelo Instituto Municipal de Urbanismo Pereira Passos, órgão da Preferitura Municipal do Rio de Janeiro responsável pela produção, armazenamento e divulgação de dados estatísticos sobre o município.

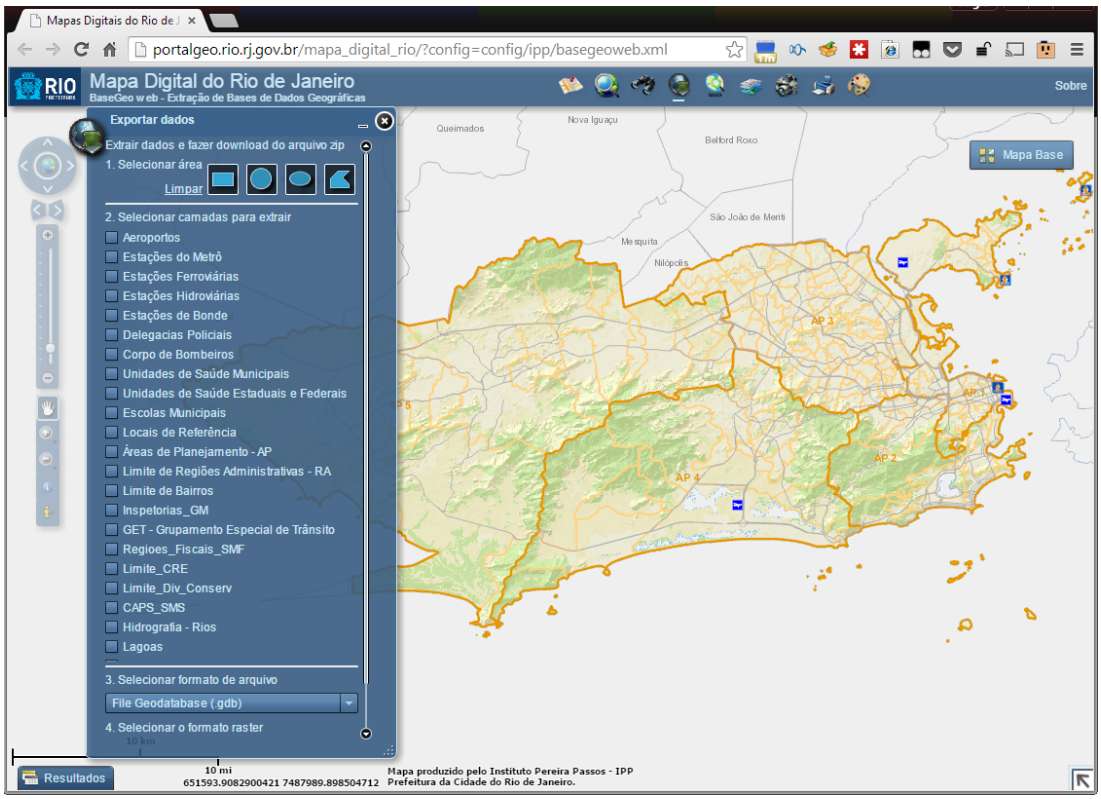


Figura 3.1: Tela Inicial do Mapa Digital do Rio de Janeiro.

O Mapa Digital do Rio de Janeiro é uma aplicação acessível pela Internet, desenvolvida na tecnologia ArcGIS da empresa ESRI, e permite ao usuário selecionar diversas camadas de informações georreferenciadas para toda a cidade ou uma área delimitada pelo usuário. Após a seleção das camadas de interesse é feito o *download*

[7]Mapa Digital do Rio de Janeiro: http://portalgeo.rio.rj.gov.br/mapa_digital_rio/?config=config/ipp/basegeoweb.xml

das informações em arquivo no formato *shapefile*, um formato proprietário da ESRI para ser utilizado em suas ferramentas, entretanto bastante popular na comunidade de Sistemas de Informações Georreferenciadas. O formato shapefile oferecido pelo Mapa Digital compõe-se de 4 arquivos distintos identificados por extensões que compartilham o mesmo nome:

arquivo .shp contém as coordenadas dos pontos, linhas e polígonos, comumente chamados de geometrias, dos elementos georreferenciados a serem projetados em uma visualização, impressão, ou utilizado em análises geoespaciais;

arquivo .shx armazena índices para otimizar operações de acesso e leitura das geometrias no arquivo .shp;

arquivo .dbf contém os atributos, informações não espaciais, dos elementos georreferenciados;

arquivo .sbn armazena índices espaciais para otimizar operações de análises georreferenciadas.

Importamos os arquivos shapefile para o banco de dados PostgreSQL utilizado no estudo com o seguinte comando no Windows:

Listing 3.1: Importar *shapefiles* para o banco de dados PostGIS.

```
for %%f in (*.shp) do
  shp2pgsql -d -W LATIN1 -I -s 29193:4326 %%f pgr_%%~nf > pgr_%%~nf.
     sql;
for %%f in (*.sql) do
  psql -d postgresql://postgres:1234@localhost/zap -f %%f;
```

O primeiro laço seleciona os arquivos de extensão **.shp** localizados na pasta em uso e cria arquivos com extenão **.sql**, e sufixo "pgr"para facilitar a identificação das tabelas criadas, com comandos SQL para a criação de tabela e cadastro dos valores. A seguir detalhamos as opções utilizadas:

-d determina sobreescrever a tabela caso ela já exista, o nome da tabela é o nome do arquivo **.shp**;

-W código UNICODE do texto dos atributos no arquivo .dbf, em nosso caso LATIN1;

-I cria índice espacial para otimizar análises geográficas;

-s transformar SRID, Identificador do Sistema de Referência Espacial[8], em nosso caso do SRID 29193, utilizado pelo Mapa Digital, para o SRID 4326, escolhido como padrão para esse estudo.

O segundo laço executa os comandos nos arquivos .sql da pasta em uso no banco de dados definido pela opção **-d**.

Decidimos registrar apenas a menor distância para cada imóvel a cada um desses elementos com objetivo de minimizar a quantidade de informações a serem utilizadas no estudo. Tomando como exemplo Corpo de Bombeiros, após a importação *shapefile* para a tabela "pgr_corpo_de_bombeiros", o seguinte comando SQL cria uma visão materializada, uma espécie de consulta de banco de dados que fica armazenada em disco para otimizar acessos futuros, associando cada imóvel ao corpo de bombeiro mais próximo e a distância entre eles:

Listing 3.2: Exemplo de cálculo de menor distância por SQL.

```
CREATE MATERIALIZED VIEW vw_dist_bombeiro AS
SELECT DISTINCT ON (s.id)   s.id, h.gid, h.nome, ST_Distance(s.geom,
    h.geom, true)
  FROM vw_imovel s
  JOIN pgr_corpo_de_bombeiros h ON ST_DWithin(s.geom, h.geom, 9999)
ORDER BY 1, 4
```

Delegacias, bombeiros, estações de trem e metrô são georreferenciados como pontos e a distância calculada é a distância entre eles e o imóvel. Os principais logradouros são georreferenciados como múltiplas linhas, sendo assim a distância calculada é a

[8]Do termo em inglês *Spatial Reference System Identifier*, tradução nossa.

tangente da linha mais próxima do logradouro ao imóvel. Por último, as favelas, lagoas são georreferenciadas como múltiplos polígonos, sendo a distância calculada a tangente do contorno mais próximo desses elementos ao imóvel.

Estabelecimentos de Saúde

As localizações dos estabelecimentos de saúde foram coletadas do Portal de Dados Abertos da Prefeitura do Rio de Janeiro[9]. Esse portal disponibiliza várias informações de interesse público a respeito da cidade do Rio de Janeiro em formato tabular CSV[10] já com o SRID 4326.

As informações dos estabelecimentos de saúde foram importados para o banco de dados seguindo as instruções em Python abaixo:

```
1  import zap_util as z
2  # Importa arquivo CSV para um dataframe.
3  df = z.pd.read_csv('../gis/data.rio/Estabelecimentos␣de␣Saude␣-␣
       Dados.csv',  encoding='iso-8859-1',  dtype=str)
4
5  # Remove acentos dos nomes das colunas e e todos os valores.
6  df.columns =  z.remove_acento(list(df.columns.values))
7  df.applymap(z.remove_acento)
8
9  # Salva dataframe no banco de dados.
10 z.d.salva_dataframe(df, '_estabelecimento_saude', index=False)
11
12 # Cria campo para armazenar a coordenada geografica nativamente.
13 z.d.__executar('ALTER␣TABLE␣_estabelecimento_saude␣ADD␣COLUMN␣␣geom␣
       geometry(Point,4326);')
14
```

[9] Portal de Dados Abertos da Prefeitura do Rio de Janeiro: http://data.rio.rj.gov.br/
[10] *Comma-separated values*: http://en.wikipedia.org/wiki/Comma-separated_values

```
15 #Cria a geometria com base na Latitude e Longitude.
16 z.d.__executar('update␣_estabelecimento_saude␣set␣geom␣=␣' + \
17     '␣st_geomfromtext(\'POINT(\'||"Longitude"||\'␣\'||"Latitude
       "||\')\',␣4326)')
18
19 # Cria chave primaria para a tabela.
20 z.d.__executar('ALTER␣TABLE␣_estabelecimento_saude' +
21   '␣ADD␣CONSTRAINT␣pk__estabelecimento_saude␣PRIMARY␣KEY("CNES");' )
```

O módulo "zap_util"importado na linha 1 contém diversas funções criadas para esse estudo e encontra-se listado no appendix A. Para esse estudo separamos os estabelecimentos em dois conjuntos, administração pública e privada, e calculamos a distância dos imóveis a cada um destes conjuntos a partir da geometria do tipo ponto criada com as informações de latitude e longitude. Os demais dados presentes no arquivo não foram utilizados.

Praias

A geometria das praias que circundam a cidade do Rio de Janeiro foram capturadas do serviço Open Street Map[11], um serviço colaborativo de registro e consulta de informações georreferenciadas, similar ao Google Maps, mas cuja licença de uso *Open Database License* garante acesso gratuito para compartilhar, modificar e usar os dados disponíveis livremente.

A obtenção dos dados foi facilitada como uso do *plugin* QuickOSM para o software Quantum GIS. Esse *plugin* permite fazer o *download* das informações disponíveis no Open Street Map que intersectem a área definida por retângulo de coordenadas personalizadas. Em nosso caso, utilizamos *software* a geometria da cidade do Rio de Janeiro para determinar as coordenadas de um retângulo circunscrito à cidade. Essas coordenadas foram passadas ao plugin, em conjunto com um filtro para obter somente

[11] Open Street Map: https://www.openstreetmap.org.

informações sobre praias. O plugin faz o download das informações pertinentes em um arquivo temporário. Como retângulo circunscrito à cidade traz informações de outras cidades vizinhas, filtramos as praias de interesse realizando nova análise espacial, desta vez no *software* Quantum GIS, para selecionar somente geometrias das praias contidas dentro da área definida pela geometria da cidade do Rio de Janeiro. Por fim, essa seleção é salva no formato *shapefile* e importadas para o banco de dados conforme descrito na listagem 3.1. O cálculo da distância é similar ao exemplo apresentado na listagem 3.2.

Centro da cidade do Rio de Janeiro

Definimos arbitrariamente o centro da cidade como a localização geográfica da estação de metrô Carioca, na Avenida Rio Branco, cujas coordenadas são em função da proximidade a essa avenida que agrega um grande número de escritórios, centros médicos e repartições publicas ao longo de seu entorno, fig. 3.2. Construímos 3 distâncias a partir deste ponto para cada imóvel: uma distância euclidiana até a localização definida, a distância até a localização definida somente na dimensão da latitude e a última somente na dimensão da longitude. Com essas duas últimas esperamos poder verificar a contribuição da distância em relação aos eixos Norte-Sul e Leste-Oeste para os preços dos imóveis.

colocar as coordenadas do centro aqui

3.3 Tratamento dos dados

O tratamento dos dados é o passo necessário para a garantir da qualidade dos dados de forma a que o modelo a ser proposto possa ser construído dentro das condições ideias, evitando perturbações não conhecidas por erros, inconsistência ou presença de valores muito além do limite esperado.

Referência para outliers?

A seguir plotamos a distribuição de preços encontrados no conjunto de dados capturado.

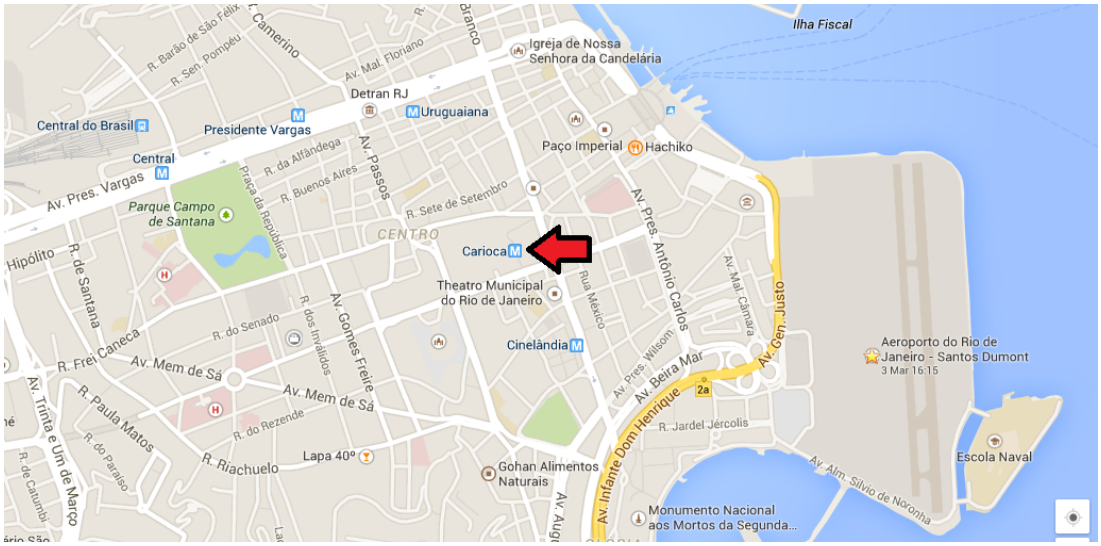


Figura 3.2: Localização da Estação de Metrô Carioca, definido como o centro da cidade.

Como os dados da pesquisa foram obtidos utilizando a técnica de Web Scrapping **??** a partir de uma origem estruturada, assume-se que

Capítulo 4

Apresentação e análise dos resultados

Em agosto e outubro de 2014 foram coletados 91.091 registros de anúncios de vendas de apartamento do tipo padrão do site ZAP Imóveis. Destes, 58.698 anúncios contém informações de localização geográfica, latitude e longitude. Em seguida, utilizando um filtro espacial para identificar aqueles cujas coordenadas geográficas estão contidos no município do Rio de Janeiro, restaram 58.353.

Teste de SVG

Após localização espacial usando as coordenadas dos anúncios, alguns bairros destacam-se por não ter nenhum anúncio:

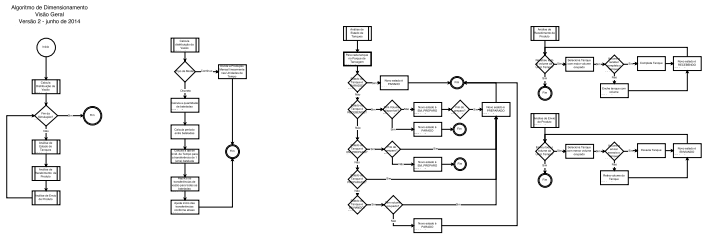


Figura 4.1

Bairro
Acari
Engenheiro Leal
Gericinó
Grumari
Mangueira
Paquetá
Parque Colúmbia
Saúde

Tabela 4.1: Bairros sem imóveis anunciados.

Em uma primeira análise, apresentamos o mapa temático abaixo, identificando os bairros pela média do m^2, para todos os imóveis anunciados.

Referências Bibliográficas

Andersen, P. K. e Skovgaard, L. T. (2010). *Regression with Linear Predictors*. Springer.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Hastie, T., Tibshirani, R., e Friedman, J. (2013). *The Elements of Statistical Learning*. Springer, 2nd edition.

Ismail, S. (2006). Spatial autocorrelation and real estate studies: A literature review.

Ismail, S. e MacGregor, B. D. (2005). Hedonic modelling of housing markets using geographical information system (gis) and spatial statistics:a case study of glasgow, scotland.

Long, F., Páez, A., e Farber, S. (2007). Spatial effects in hedonic price estimation: A case study in the city of toronto. *Working Paper Series*.

Macedo, P. B. R. (1999). Hedonic price models with spatial effects: an application to the housing market of belo horizonte, brazil. *XIV Latin American Meeting of the Econometric Society*.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, 1st edition.

Murphy, K. P. (2012). *Machine Learning, a probabilistic perspective*.

Neto, E. F. (2002). Estimação do preço hedônico: uma aplicação para o mercado imobiliário da cidade do rio de janeiro.

NIST/SEMATECH (1977). *e-Handbook of Statistical Methods*. National of Standards and Technology, 1st edition.

Apêndice A

Listagem do módulo zap_util.py

Listing A.1: Módulo zap_util.py.

```
# -*- coding: latin-1 -*-


import sys,os;
import pandas as pd
import psycopg2
from mpltools import style
import unicodedata
import geopandas as gd
from scipy.stats import norm as gauss, probplot, cumfreq
from matplotlib import rcParams
from  matplotlib.pyplot import figure, xlim, ylim,pcolor, colorbar,
   xticks, \
    yticks, subplots
from numpy import linspace, arange,std, polyfit, polyval,sqrt
```

```
x = os.getcwd()
l = x.rfind('\\')
path = x[:l+1]+'capturar_zap'
sys.path.append(path)

import dataset as d

# Connect to an existing database
con = psycopg2.connect("dbname=zap user=postgres")

def set_style(sty='ggplot'):
    style.use(sty)



def exec_sql(sql, con=None):
    if con == None: con = d.conecta_db()
    return d.__executar(sql,con=con)

def get(sql,id_='id',con=None):
    if con == None: con = d.conecta_db()
    return pd.io.sql.read_sql(sql, con, index_col=id_)

def get_geo(sql, con=None):
    if con == None: con = d.conecta_db()
    return gd.GeoDataFrame.from_postgis(sql, con)

tam_fig_original = None
def tam_figura(largura=None, altura=None):
```

```
    if tam_fig_original == None:
        tam_fig_original = rcParams['figure.figsize']

    if largura == None and altura == None:
        rcParams['figure.figsize'] = tam_fig_original
    if largura != None and altura == None:
        rcParams['figure.figsize'] = [largura*i for i in
           tam_fig_original]
    if largura != None and altura == None:
        rcParams = [largura, altura]
    if largura == None and altura != None:
        raise Exception('Ou␣ambos␣as␣variaveis␣sao␣nulas␣ou␣somente␣
           "largura"␣eh␣definida.')




def remove_acento(str_or_list):
    troca = []
    troca.append( {'de':[' ',' ',' ',' ',' ',' '], 'para':'a'})
    troca.append( {'de':[' ',' ',' ',' ',' ',' '], 'para':'A'})
    troca.append( {'de':[' ',' ',' ',' ',' '], 'para':'e'})
    troca.append( {'de':[' ',' ',' ',' ',' '], 'para':'E'})
    troca.append( {'de':[' ',' ',' ',' ',' '], 'para':'i'})
    troca.append( {'de':[' ',' ',' ',' ',' '], 'para':'I'})
    troca.append( {'de':[' ',' ',' ',' ',' ',' '], 'para':'o'})
    troca.append( {'de':[' ',' ',' ',' ',' ',' '], 'para':'O'})
    troca.append( {'de':[' ',' ',' ',' ',' '], 'para':'u'})
    troca.append( {'de':[' ',' ',' ',' ',' '], 'para':'U'})
```

```
    troca.append( {'de':[' '], 'para':'c'})
    troca.append( {'de':[' '], 'para':'C'})
    troca.append( {'de':['(', ')','[',']'], 'para':'_'})

    if type(str_or_list) == str:
        str_sem = str_or_list
        for item in troca:
            for c in item['de']:
                str_sem = str_sem.replace(c,item['para'])
        return str_sem

    if type(str_or_list) == list:
        return [remove_acento(item) for item in str_or_list]

    if type(str_or_list) == unicode:
        return unicodedata.normalize('NFKD', str_or_list).encode('
           ascii', 'ignore')


    msg =  'ERRO: ''unicode'', ''str'' ou ''list'' esperado. ' + str
       (type(str_or_list)) + ' encontrado.'
    raise Exception(msg)

def plot_residual(smres):

    resid_std = (smres.resid-smres.resid.mean())/smres.resid.std()
    fig = figure()
    w,h = rcParams['figure.figsize']
    fig.set_size_inches(w*2,h*2)
    ylim_ = (resid_std.min()*1.01,resid_std.max()*1.01)
```

```
xlim_ = (smres.fittedvalues.min(),smres.fittedvalues.max())
ax = fig.add_subplot(221)
probplot(resid_std, plot=ax);
ylim(ylim_);



ax = fig.add_subplot(222)
ax.scatter(smres.fittedvalues,resid_std);

# TODO: Parou de funcionar, substituido pelo codigo logo a
   seguir.
#ax.axhline(y=smres.resid.mean(),
#           xmin=smres.fittedvalues.min(),
#           xmax=smres.fittedvalues.max(),color='r');
x = linspace(smres.fittedvalues.min(),smres.fittedvalues.max()
   ,5)
y = [smres.resid.mean() for i in range(len(x))]
ax.plot(x,y, 'r-', linewidth=2)

xlim(xlim_);
ylim(ylim_);

ax = fig.add_subplot(223)
ax.hist(resid_std,50);

from matplotlib.colors import LogNorm
ax = fig.add_subplot(224)
ax.hexbin(smres.fittedvalues,resid_std, norm=LogNorm());
ax.axhline(y=smres.resid.mean(),xmin=smres.fittedvalues.min(),
   xmax=smres.fittedvalues.max(),color='r');
```

```
    xlim(xlim_);
    ylim(ylim_);

def prep_formula(dataframe, dataframe_name, var_dep='preco', func=
    None):




    # Determinar colunas que so identificadoers para serem
        removidos.
    cols_ids = set([c for c in dataframe.columns if c.find('id_')
        >-1])

    # Identificar colunas que contm valores ausentes.
    s = dataframe.isnull().sum()
    cols_nulos = set(s[s>0].index.tolist())

    # Colunas que so variveis dependentes.
    cols_dep = set(['m2','preco'])

    # Juntar todas as colunas a serem removidas do modelo.
    cols_excluded = cols_ids.union(cols_nulos).union(cols_dep)

    # Definir as variveis para o modelo.
    cols = set(dataframe.columns) - cols_excluded

    # Construir a frmula.
    if func != None:
        yname = '{}({}.{}) '.format(func, dataframe_name, var_dep)
    else:
```

```
        yname = '{}.{} '.format(dataframe_name, var_dep)
    formula = yname + ' ~ ' +  \
        ' + '.join([dataframe_name + '.'+c for c in cols])

    return formula, cols, cols_excluded

def plot_corrmatrix(matrix, figsize=(8*1.05, 6*1.05),**args):

    figure(figsize=figsize)
    pcolor(matrix,**args);
    colorbar();

    max_ = len(matrix);
    yticks(arange(0.5,max_+0.5), range(0,max_));
    xticks(arange(0.5,max_+0.5), range(0,max_));



def cols_autocorr(matrix, threshold = 0.69999):
    from numpy import tril
    matrix.loc[:,:] =  tril(matrix, k=-1) # borrowed from Karl D's
       answer

    already_in = set()
    result = {}
    for col in matrix:
        perfect_corr = matrix[col][abs(matrix[col]) > threshold].
           index.tolist()
        #if perfect_corr and col not in already_in:
        #    already_in.update(set(perfect_corr))
        #    perfect_corr.append(col)
```

```
        #    cols_autocorr.append(perfect_corr)
        if len(perfect_corr) > 0:
            result[col] = [i+'({:.2f})'.format(matrix[col][i]) for i
                in perfect_corr]

    return result

def print_autocorr(dataframe,cols_excluded=[]):
    if type(cols_excluded) != set:
        cols_excluded = set(cols_excluded)
    # Selecionar as colunas do modelo.
    valid_cols  = set(dataframe.columns.tolist()) - cols_excluded
    valid_cols = list(valid_cols)



    matrix_corr = dataframe[valid_cols].corr()

    cols_autoc = cols_autocorr(matrix_corr)

    if len(cols_autoc) == 0:
        print 'N o␣h ␣colunas␣autocorrelacionadas.'
    else:
        print 'Coluna'.ljust(20),'|', 'Autocorrelacionada␣com␣'.
            ljust(50)
        for k,c in cols_autoc.iteritems():
            print str(k).ljust(20),':', str(c).ljust(50)

def scatter_distancia(dfx,suptitle=None):
    # Colunas que representam distncias.
    dist_columns = sorted([c for c in dfx.columns if c.find('dist_')
```

```
        > -1])


    # Definir tamanho do grfico.
    w,h = rcParams['figure.figsize']
    f,a = subplots(len(dist_columns), 2)
    f.set_size_inches(w*2, h*len(dist_columns))


    # Ttulo central da figura.
    if suptitle != None:
        f.suptitle(suptitle)

    # Plotar grficos.
    for i in range(len(dist_columns)):
        col_name = dist_columns[i]
        x = dfx[col_name]
        fx = linspace(x.min(),x.max(),50)

        a[i, 0].scatter(x, dfx.preco)
        p = polyfit(x,dfx.preco,1)
        a[i, 0].plot(fx, polyval(p,fx), linewidth=2)
        a[i, 0].set_title(col_name + u' x pre o')

        a[i, 1].scatter(x,dfx.m2)
        p = polyfit(x,dfx.m2,1)
        a[i, 1].plot(fx, polyval(p,fx), linewidth=2)
        a[i, 1].set_title(col_name + u' x $R\$/m^2$')
```

```
def plot_boxhist(x,titulo=None,xlabel=None):
    f,a = subplots(2,1)
    a0,a1 = a.ravel()
    bp = a0.boxplot(x,vert=False);
    a0.text(max(x)*0.8,1.2,
        s='$3\sigma$={:.2f}'.format(3*std(x)),
        bbox={'facecolor':'w', 'pad':10, 'alpha':0.5},
        style='italic',fontsize=15)

    a1.hist(x,bins=30);

    if titulo != None:
        a0.set_title(titulo);
    if xlabel != None:
        a1.set_xlabel(xlabel);



def rmse(resid):
    return sqrt((resid**2/len(resid)).sum())
```

Apêndice B

Listagem de outra coisa.