Capítulo 2

Métodos Experimentais em Interação Humano Computador

Carlos H. Morimoto e Antonio Diaz-Tula

Abstract

In this course you will learn the basics to conduct scientific research in Human Computer Interaction (HCI), i.e., to plan, conduct, and analyze the results of an experiment with users interacting with some computational device. These experiments are fundamental for the innovation and development of interactive products. The course blends theory and practice, beginning with a brief introduction to HCI and the interaction design process to motivate the need to conduct user experiments. To conduct an experiment, you will be required to define your methodology and follow a rigorous statistical analysis to validate your results. At the end you will conduct a simple experiment to apply these concepts and evaluate a model that helps predict user performance in pointing tasks.

Resumo

Nesse curso você vai aprender alguns fundamentos para conduzir pesquisa científica na área de Interação Humano Computador (IHC), ou seja, como planejar, conduzir e analisar os resultados de um experimento com usuários de algum sistema computacional interativo. Esses experimentos são fundamentais para a inovação e desenvolvimento de produtos interativos. O curso alia teoria e prática, começando com uma breve introdução da área para motivar a necessidade da realização de experimentos com usuários. A condução de experimentos requer a definição da metodologia e uma rigorosa análise estatística para comprovação dos resultados. Ao final, você vai conduzir um experimento para aplicar esses conceitos e avaliar um modelo que ajuda a prever o desempenho das pessoas em tarefas de apontamento.

2.1. Introdução

Os computadores de uso genérico, até a década de 70, eram máquinas grandes, caras, a que poucas pessoas tinham acesso. Eram tipicamente usadas por instituições de grande

e médio porte, tanto públicas quanto privadas, para processar seus dados. As poucas pessoas que tinham contato e acesso a essas máquinas costumavam ser altamente treinadas e com muitos anos de experiência e, assim, não havia grande preocupação com a ”interação” com computadores. Apenas a partir da década de 80, com o surgimento dos microcomputadores pessoais de mesa, que o problema de interação começou a ganhar evidência.

A área de Interação Humano Computador (IHC) floresceu dessa necessidade de desenvolver melhores computadores. Não apenas máquinas mais velozes, poderosas e baratas. Mas melhores no sentido de serem mais úteis, mais fáceis de aprender e eficazes para as pessoas que fazem uso do computador em pequenos estabelecimentos comercias e em seus lares. Essas pessoas usuárias de microcomputadores, principalmente nas décadas de 80 e 90, não tinham conhecimento nem experiência prévia com o uso de computadores. As conquistas realizadas na tentativa de vencer tal desafio, de tornar uma máquina tão complexa mais adequada às necessidades das pessoas que tentavam usá-la, é uma das razões que torna a área de IHC tão fascinante pois a sua própria evolução está diretamente relacionada à dramática alteração nas práticas de desenvolvimento de sistemas computacionais interativos.

Uma dessas práticas é o envolvimento das próprias pessoas usuárias no processo de desenvolvimento, como no método de **Design Centrado no Usuário** (*User-Centered Design*) [Sharp et al. 2019]. A Figura 2.1 ilustra esse processo iterativo. O processo tem início com um entendimento dos **requisitos** do sistema para satisfazer as necessidades das pessoas usuárias. A etapa de **desenho** busca desenvolver soluções que pode requerer um maior entendimento sobre as pessoas e algumas ideias podem ser rapidamente avaliadas. As melhores ideias devem ser prototipadas para uma avaliação mais rigorosa. Como o resultado da avaliação pode revelar problemas, dependendo da origem e severidade desses problemas, todas as etapas anteriores podem ser revistas. O processo se encerra quando a avaliação revelar que o projeto satisfaz os requisitos.

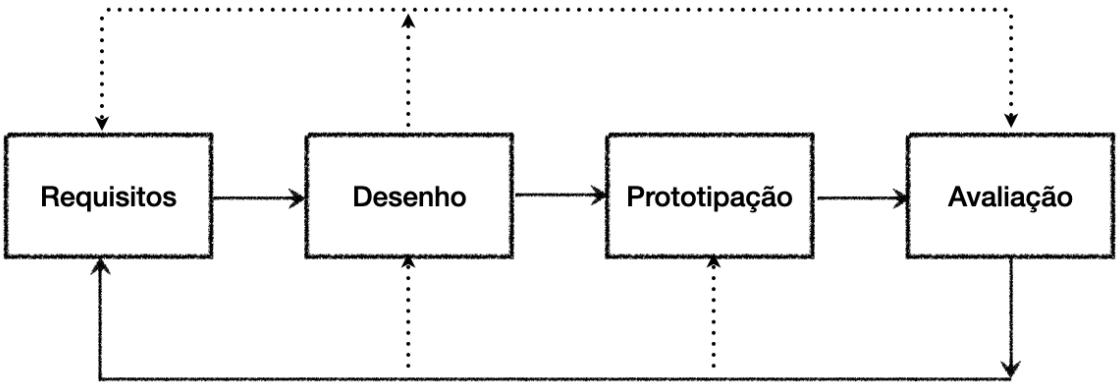

Figura 2.1. Processo de Design Centrado no Usuário.

O uso dessa prática é fundamental para evitar um foco excessivo na ”engenharia” do sistema. Um projeto focado apenas na engenharia também resulta em produtos funcionais. No entanto, o processo de Design Centrado no Usuário estende o foco para a forma e/ou maneira de usar. Por fim, o envolvimento de pessoas usuárias garante que

a forma de usar é adequada e realmente satisfatória aos usuários (ao invés de adequado aos engenheiros). Historicamente, muitos projetos desenvolvidos com foco na engenharia fracassam por não satisfazerem os usuários finais e descobrirem muito tardiamente os motivos dessas insatisfações. Essas novas práticas facilitam as descobertas desses problemas com antecedência e portanto aumentam as chances de sucesso de um projeto.

Muitas variações desse processo foram introduzidas ao longo dos anos e que, tipicamente, variam o nível de participação que as pessoas usuárias tem sobre a definição do projeto e durante as etapas do projeto. Por exemplo, o método de *Design Thinking* usado pela Escola de projeto da Universidade de Stanford (https://dschool.stanford.edu/resources) busca conhecer seu público alvo profundamente, com um alto grau de empatia, para desenvolver soluções inovadoras que satisfaçam alguma necessidade específica. Observe que esse processo de desenho é genérico e não requer que o produto seja computacional.

Esse processo se baseia em testar as ideais rapidamente, envolvendo usuários. A prototipação permite que as pessoas possam experimentar e criticar ideias fundamentais do projeto antes de serem concretizadas na forma de produto. Esse processo iterativo de projeto, com avaliações frequentes, leva a criação de produtos realmente úteis e satisfatórios.

Boa parte das avaliações, principalmente nas fases iniciais de um projeto, podem utilizar **métodos de desconto** (*discount methods*), como o método de avaliação heurística de usabilidade proposta por Nielsen [Nielsen 1994]. Essas avaliações são ágeis, simples de executar e de baixo custo, mas por terem propósitos específicos, o escopo dos resultados dessas avaliações tende a ser limitado ao produto sendo desenvolvido.

A pesquisa (científica) em IHC possui objetivos mais fundamentais, que não estão voltados a um produto ou projeto específico e, por isso, seus resultados podem ser generalizados e utilizados para melhorar sistemas existentes e também podem contribuir no desenvolvimento de novos sistemas interativos inovadores. Um exemplo específico desse processo é o iPhone lançado em 2007 que, segundo o Prof. Selker [Selker 2008], ”por meio do iPhone, a Apple conseguiu reunir com sucesso décadas de pesquisa em um só produto”, como as pesquisas realizadas sobre gestos usando múltiplos toques realizadas desde a década de 80 [Buxton et al. 1985].

A pesquisa em IHC tipicamente depende de comprovação empírica que confirma (ou refuta) uma ideia. Para isso as ideias precisam ser colocadas na forma de hipóteses cuja validade deve ser testada em experimentos com usuários. Na próxima seção vamos discutir os tipos, as razões e as maneiras como esses experimentos podem ser realizados. Antes porém, queremos descrever melhor nossa motivação ao propor esse curso para ser oferecido no JAI 2021.

2.1.1. Motivação e objetivos

Esse curso surgiu do interesse de alguns alunos e alunas durante aulas da disciplina MAC0446 (”Princípios de Interação Humano Computador”) oferecida pelo Departamento de Ciência da Computação (DCC) do Instituto de Matemática e Estatística (IME) da Universidade de São Paulo (USP) de aprender mais sobre o desenho de experimentos com

usuários e sobre a análise dos seus resultados.

Esse curso, portanto é voltado para alunas e alunos de graduação e talvez no início de uma pós-graduação, que já conheçam um pouco da área de IHC e que tenham conhecimentos básicos de computação e estatística. No entanto, embora nossa motivação inicial tenha sido IHC, gostaríamos que esse curso no JAI possa acolher pessoas de outras áreas, fora de IHC e até fora da computação, que tenham interesse de realizar estudos com usuários e/ou aprender um pouco sobre o método científico de pesquisa.

2.1.2. Organização do curso

Por ser um curso de poucas horas, nenhum assunto será coberto com a devida profundidade. Nosso principal objetivo é introduzir e discutir alguns conceitos a partir de um exemplo real baseado em um artigo publicado pelos autores do curso e fomentar a curiosidade científica dos participantes para que inventem e realizem seus próprios experimentos, apoiados pela grande variedade de ferramentas estatísticas e computacionais que existem hoje à disposição.

A Seção 2.2 desse texto resume vários capítulos do livro *Human Computer Interaction: An Empirical Research Perspective*, do Prof. Scott Mackenzie [Mackenzie 2012]. Caso você tenha interesse, recomendamos a leitura desse livro para continuar o seu aprendizado sobre métodos experimentais em IHC. Como sugestão de outros livros mais recentes, que cobrem as técnicas de análise estatística com um pouco mais de profundidade, citamos também:

- *Research Methods in Human Computer Interaction*, de Lazar, Feng e Hocheiser [Lazar et al. 2017]; e
- *Modern Statistical Methods for HCI*, por Robertson e Kaptein (editores) [Robertson and Kaptein 2018].

Além do conhecimento estatístico, a exploração dos dados requer o uso de boas ferramentas computacionais. Nós escolhemos usar ferramentas baseadas na linguagem Python devido à generalidade da linguagem e sintaxe simples, que facilita o aprendizado e entendimento dos trechos de código utilizados nesse curso.

Ao final do curso, vamos sugerir um experimento que você pode fazer em sua própria casa para aplicar esses conceitos e ferramentas. Trata-se de um típico experimento para avaliar o desempenho humano em uma tarefa simples, para levantar uma curva de desempenho conhecida como lei de Fitts [Fitts 1954].

2.2. Pesquisa em IHC

Quando dizemos que uma pesquisa é empírica significa que seus resultados provêm de um experimento envolvendo pessoas usuárias realizando uma determinada tarefa. Mas nem todas as pesquisas envolvem experimento. Muito se pode descobrir sobre as pessoas usuárias por meio de uma pesquisa de opinião por exemplo, onde as pessoas são consultadas diretamente (por meio de uma entrevista ou respondendo a um questionário) sobre como utilizam um determinado sistema ou qual sistema elas utilizam com mais

frequência. Além disso, algumas pesquisas desse tipo podem ser feitas por meio de consultas ”indiretas” baseadas no número de cliques ou visitas de uma página na Internet. Esse método de pesquisa é conhecido por **método observacional** e é muito comum em ciências sociais. Seus resultados tendem a ser mais **qualitativos** que **quantitativos** e, por isso, são mais utilizados para descobrir a razão (o ”por quê”) ou a maneira que (o ”como”) as pessoas realizam a interação.

Nessa seção vamos discutir como realizar pesquisas segundo o **método experimental**, também conhecido como **método científico**, onde o conhecimento é adquirido por meio de **experimentos controlados**. No caso de IHC, os experimentos tipicamente envolvem pessoas e por isso é costumeiramente chamado de **estudo com usuários** (*user study*). O método científico procura responder uma questão de pesquisa cuja resposta depende da comprovação de uma hipótese. Essa hipótese é testada em um experimento e o resultado é analisado estatisticamente para determinar a confiança que podemos ter no resultado.

Uma importante característica do método científico é que o experimento deve ser reproduzível, ou seja, ao realizar o mesmo experimento várias vezes, devemos chegar a mesma conclusão. Por isso, as condições do experimento tendem a ser controladas, o que pode reduzir de certa forma a relevância dos resultados em IHC pois as pessoas realizam as tarefas em condições artificiais (controladas).

Um experimento controlado requer ao menos duas variáveis: uma variável a ser **manipulada** e outra a ser medida para indicar a **resposta**. Em IHC, a variável manipulada em geral corresponde a um parâmetro modificável na interface, como o tamanho de uma tecla virtual em um editor de texto. Já a resposta pode ser uma medida qualitativa como a opinião dos participantes sobre o conforto oferecido pelo teclado ou uma medida quantitativa como a velocidade ou taxa de erros de digitação.

2.2.1. AugKey: exemplo de um estudo com usuários

Para contextualizar nossos exemplos e facilitar o entendimento dos conceitos que vamos apresentar a respeito de estudos com usuários vamos utilizar um experimento real publicado pelos autores utilizando um teclado virtual com predição (denominado AugKey [Diaz-Tula and Morimoto 2016]). O objetivo do AugKey é melhorar a experiência de pessoas com deficiências motoras na tarefa de entrada de texto utilizando apenas o olhar.

Para realizar a entrada de texto (”digitação”) usando apenas os movimentos dos olhos utilizamos um dispositivo rastreador de olhar como ilustrado na Figura 2.2. O rastreador de olhar [Morimoto and Mimica 2005] é tipicamente composto por uma câmera (colocada sob o monitor) que rastreia os movimentos dos olhos e os mapeia para a tela do próprio computador, como mostrado na figura. Esse dispositivo permite, por exemplo, controlar o cursor na tela pelo olhar, ao invés de usar o mouse ou outro dispositivo de apontamento manual. O custo desses dispositivos tem caído muito e atualmente há modelos comerciais de rastreadores de olhar que podem ser acoplados diretamente à tela de monitores de mesa e notebooks, usados por exemplo em alguns jogos eletrônicos.

Um uso tradicional de rastreadores de olhar é na área de acessibilidade. Algu-

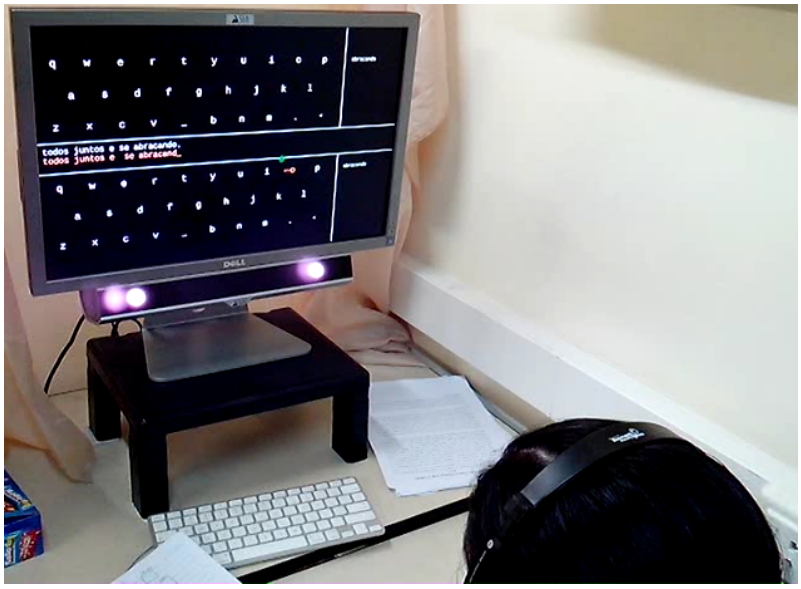

Figura 2.2. Entrada de texto pelo olhar usando um rastreador remoto.

mas pessoas com deficiências motoras severas, como por exemplo pessoas afetadas por Esclerose Lateral Amiotrófica ou Síndrome do Encarceramento, não possuem ou perdem controle muscular dos membros do corpo e, portanto, não conseguem utilizar dispositivos convencionais como teclado e mouse. Muitas dessas pessoas se beneficiam de aplicativos controlados pelo olhar para interagir com o computador e, em casos mais severos quando a pessoa não tem controle da fala, essas interfaces se tornam também um meio importante para elas se comunicarem. A entrada de texto pelo olhar, além de facilitar a comunicação dessas pessoas, permite a sua integração digital com a sociedade.

Tipicamente, a digitação pelo olhar é realizada utilizando um teclado virtual exibido no monitor, como ilustrado na Figura 2.3. O primeiro desafio é como ”clicar” em uma tecla. Algumas pessoas conseguem usar um botão acionado mecanicamente usando a boca ou o pé, ou ainda piscando os olhos ou balançando a cabeça. Uma solução frequentemente utilizada e baseada apenas no olhar é usar tempo de latência (*dwell-time*) para acionar a tecla sendo focada. Assim, quando a usuária deseja digitar uma tecla, ela deve olhar para a tecla e permanecer olhando por um determinado tempo, sem desviar o olhar da tecla, até ela ser acionada. Uma ampulheta (ícone que indica a passagem do tempo) é exibida para que a pessoa saiba quanto tempo ela precisa esperar e/ou até quando ela pode desistir de olhar para evitar uma seleção indesejada. É comum utilizar tempos de latência entre 500 e 1000 ms.

Digitar por meio do olhar é um processo lento e pode causar fadiga visual após um período mais prolongado. Para reduzir a fadiga visual e aumentar a velocidade de digitação, podemos utilizar a técnica de predição de palavras encontrada em vários editores de texto, em particular, aqueles utilizados em dispositivos móveis. A predição de palavras permite que o sistema ofereça algumas alternativas logo após a digitação de alguns caracteres. A palavra candidata mais provável pode aparecer como opção de *autocompletar*, e mais palavras candidatas podem aparecer em outra região da interface. Para utilizar uma dessas outras palavras, basta a pessoa clicar na opção desejada na lista de

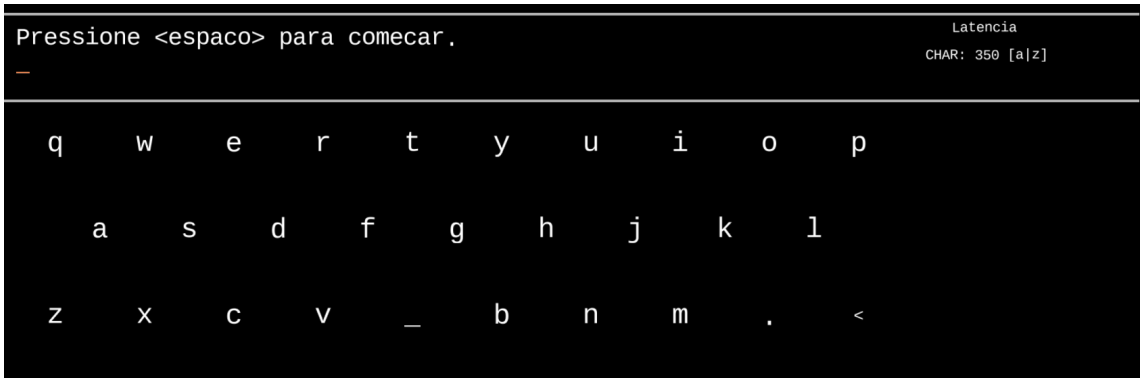

Figura 2.3. Teclado virtual para entrada de texto pelo olhar. Os movimentos oculares são rastreados usando uma câmera de vídeos apontada para os olhos.

palavras candidatas [Majaranta 2009, Wobbrock et al. 2008, Urbina and Huckauf 2010].

Embora o uso de predição de palavras tenha o potencial de aumentar a velocidade de digitação pelo fato de economizar a digitação dos caracteres que são completados, a necessidade de alternar o foco entre o teclado sendo acionado pelo olhar e a lista de palavras candidatas exibidas em outra região da tela limita bastante o ganho obtido na prática. Além disso, a pessoa usuária precisa olhar repetidamente para a área de texto para detectar erros na digitação e confirmar o que realmente foi digitado.

Com o objetivo de aliviar o problema de desviar o foco repetidamente para a lista de palavras, propomos em [Diaz-Tula and Morimoto 2016] um novo modelo de interação chamado de *AugKey* (*Augmented Keys*). Nossa ideia foi aumentar a quantidade de informação exibida na tecla sob o foco do olhar para que essas informações possam ser consumidas durante a latência, sem que o olhar precise se mover. Assim o olhar não precisaria mais ser deslocado da tecla focada para procurar informação sobre o texto que acabou de ser digitado, pois essa informação já se encontra na área da tecla focada. A Figura 2.4 mostra um teclado virtual com AugKey. Observe que, com AugKey, dentro da região da tecla e ao redor do caractere recebendo o foco do olhar (a tecla ”t” na figura) são mostradas informações adicionais: os últimos caracteres digitados e as palavras candidatas que serão exibidas na lista de palavras candidatas após a seleção do caractere focado (fim do tempo de latência).

As perguntas de pesquisa que devemos investigar devem comprovar se as alterações propostas pelo AugKey tem algum efeito sobre os teclados tradicionais e, caso tenha algum efeito, como podemos medir se esse efeito é positivo ou negativo. Por exemplo, desejamos saber qual método permite atingir uma maior velocidade de digitação, ou qual deles é mais confortável, ou mais fácil de aprender. Ao projetar um experimento devemos definir as variáveis que devem ser manipuladas para que outras possam ser medidas, como veremos a seguir.

2.2.2. Variáveis e Dados

A sofisticação das conclusões do experimento estão diretamente relacionadas às medidas e, por isso, para se obter conclusões mais elaboradas é necessário utilizar medidas quantitativas sempre que possível. No entanto, há dados de natureza qualitativa que são

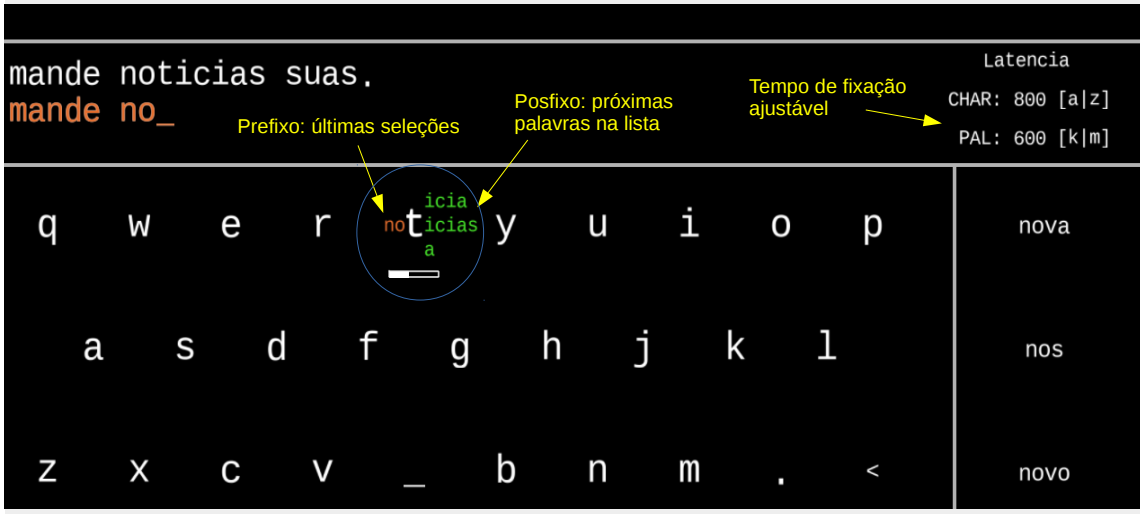

Figura 2.4. Teclado virtual controlado por tempo de latência com uma lista de predição de palavras baseado em AugKey. A lista de palavras é mostrada do lado direito da interface. A tecla recebendo o foco do olhar exibe os últimos caracteres digitados, as próximas palavras a serem apresentadas na lista de predições, e a ampulheta sob o caractere.

essenciais também para avançar o conhecimento. Cada tipo de dado exige um tratamento diferente para ser analisado.

Além do tipo, devemos considerar também a escala de medição utilizada. É comum utilizar as seguintes quatro escalas: nominal, ordinal, intervalar e proporcional. Escalas nominais e ordinais são tipicamente usadas para medir dados **qualitativos** enquanto escalas intervalares e proporcionais são usadas para medir dados **quantitativos**. Vamos a seguir descrever propriedades dessas escalas para entender seus usos e limitações.

2.2.2.1. Escala Nominal

Um dado em escala nominal (dado nominal) é basicamente atribuído a uma categoria como, por exemplo, sexo (masculino ou feminino), nacionalidade (brasileira, cubana, japonesa etc.), ou tipo de animal preferido de estimação (cachorro ou gato). Essa escala é usada para agrupar os dados em classes ou categorias distintas.

Dados em escala nominal limitam manipulações matemáticas e estatísticas. Por exemplo, será que podemos calcular a média dentre os animais de estimação para obter um bicho metade cachorro e metade gato?

Dados nominais são tipicamente usados para exibir a frequência ou número de ocorrências em cada categoria, como por exemplo, dizer coisas como: 72% dos moradores do bairro Periferia tem animais de estimação, desses 67% são cães, 21% gatos e 12% possuem outros animais como roedores, tartarugas e pássaros.

No experimento do AugKey a variável ”teclado” é de tipo nominal e possui dois valores possíveis como ”Sem predição” de palavras e ”AugKey”.

2.2.2.2. Escala Ordinal

Um dado em escala ordinal (dado ordinal) permite que ele seja ordenado segundo alguma propriedade. Por exemplo, podemos entrevistar algumas pessoas e pedir para que elas ordenem três cores, ou perfumes, ou marcas de refrigerante, segundo sua preferência. Assim, certa pessoa pode ter como fruta preferida ”melancia”, seguido de ”manga” e ”mamão”. Ou ainda, podemos perguntar a uma pessoa que ordene propriedades que ela considera ao comprar algum produto como preço, beleza, durabilidade e conforto. Como resultado de uma pesquisa usando essas propriedades considerando, por exemplo, um sofá, podemos descobrir que a beleza é o ponto mais importante para nossos ”clientes” (pessoas que participaram da pesquisa e potenciais usuárias de sofás), seguido de preço, durabilidade e, por último, conforto.

A principal limitação do uso de dados ordinais é que geralmente a diferença entre pontos sucessivos na escala não é a mesma, ou seja, a diferença entre o preço e a durabilidade pode ser grande para os clientes, enquanto a diferença entre durabilidade e conforto pode ser pequena.

Observe que ainda não faz sentido calcular a média desses dados mas eles fornecem uma informação mais rica que dados nominais pois permitem comparações do tipo maior e menor, como *pessoas consideram preço mais relevante que conforto*, nesse exemplo.

Voltando ao AugKey, há várias propriedades qualitativas que poderiam ser ordenadas, como por exemplo o conforto oferecido por cada interface ao digitar e a preferência de cada pessoa por um dos métodos. Além disso, mesmo propriedades quantitativas, como velocidade de digitação e número de erros, podem receber ordenações segundo a percepção das usuárias. É possível, por exemplo, que a velocidade de digitação de um método seja percebida como menor que outro método embora, ao ser medida quantitativamente, se revele maior.

2.2.2.3. Escala Intervalar

Um dado em escala intervalar (dado intervalar) apresenta distâncias iguais entre valores adjacentes. No entanto, a escala ainda limita certas comparações pela falta de uma referência absoluta (um zero). Um exemplo típico de dado intervalar (e que pode confundir você) é a temperatura medida em graus Celsius.

Dados intervalares, como temperatura, permitem o cálculo de medidas estatísticas como a média e a variância da temperatura em um determinado dia do ano. No entanto, não é válido considerar o resultado de proporções desses dados. Por exemplo, não deveríamos dizer que 30^oC é seis vezes mais quente que 5^oC (ou ainda menos três vezes mais quente que -10^oC?).

Em IHC é comum também considerar os dados de questionários com respostas na forma de escalas de Likert como dados intervalares. Embora exista evidência de que as pessoas percebam os itens nos extremos da escala como mais afastados que os itens no centro [Kaptein et al. 2010], para que a média dos valores faça sentido é necessário con-

siderar que os intervalos possuam a mesma distância. Uma solução é instruir as pessoas participantes a considerar que as categorias possuem mesma distância.

Observe que dados intervalares são mais ricos que dados ordinais e que é possível transformar esses dados em ordinais e até nominais, por exemplo, classificando temperaturas nas categorias ”quente” e ”frio”, ou ainda ”quente”, ”morno” e ”frio” etc. No caso de teclados, a velocidade de digitação pode ser percebida como teclado ”rápido” ou ”lento”.

2.2.2.4. Escala Proporcional

Um dado em escala proporcional (dado proporcional) permite o cálculo de proporções, ou razões entre valores (por isso é também chamada de escala de razão (*ratio scale*)) pois possuem um zero absoluto. Tais medidas permitem o cálculo de uma série de medidas que contribuem para conclusões mais elaboradas sobre os resultados do experimento. Esses dados podem ser somados e subtraídos, multiplicados e divididos, para calcular médias, desvios e variâncias estatísticas.

Em IHC, a medida mais comum utilizada em escala proporcional é o tempo, na forma de intervalo de tempo utilizado para completar uma tarefa. Até mesmo a idade e anos de experiência (tempo) são dados proporcionais bastante utilizados. Podemos assim dizer que certa pessoa possui duas vezes mais experiência que outra.

Em geral, qualquer outra medida física além de tempo, como a força que uma pessoa deve exercer sobre um controle ou é percebida de um dispositivo háptico, ou a distância percorrida pelo cursor no monitor, são também medidas proporcionais.

No exemplo que estamos estudando sobre comparação de teclados virtuais, a velocidade de digitação medida em número de caracteres (ou palavras) por minuto representa uma variável em escala proporcional.

2.2.3. Metodologia

Enquanto o método científico define o processo, a metodologia descrita nessa seção se refere ao ”estudo” desse processo que define como o experimento deve ser realizado para responder à pergunta da pesquisa. No caso de IHC, isso envolve a escolha dos participantes, os materiais (hardware e software), as tarefas (e como elas devem ser realizadas), as instruções para receber e treinar os participantes antes do experimento e coletar informações individuais durante e após seu término, as variáveis a serem manipuladas e medidas, a forma de coleta e análise dos dados etc.

Definir uma metodologia adequada é fundamental para que possamos confiar nos resultados e replicá-los. Uma metodologia deficiente não permite conclusões (pois impossibilita sua replicação) ou aumenta a incerteza sobre os resultados (os coloca em dúvida).

Por envolver pessoas, é muito importante também submeter o experimento a um comitê de ética e só iniciar o experimento após a sua aprovação. No Brasil, é necessário submeter seu projeto pela Plataforma Brasil [http://plataformabrasil.saude.gov.br], que encaminha o projeto a um comitê de ética próximo ao local onde o projeto será conduzido.

A Plataforma Brasil é uma base nacional e unificada de registros de pesquisas envolvendo seres humanos mantida pelo Ministério da Saúde. Ela permite que as pesquisas sejam acompanhadas em seus diferentes estágios - desde sua submissão até a aprovação final pelo comitê de ética que acompanha o projeto. O sistema permite a apresentação de documentos também em meio digital, propiciando à sociedade o acesso aos dados públicos de todas as pesquisas aprovadas.

O comprometimento ético requer também que os participantes sejam esclarecidos sobre:

- a natureza e os propósitos da pesquisa;
- a metodologia utilizada (procedimentos, questionários etc);
- os riscos e benefícios da participação;
- o direito de não participar, interromper a participação a qualquer momento e de não precisar responder as perguntas que não desejar, sem incorrer em nenhum prejuízo ou penalidade;
- o direito ao anonimato e confidencialidade dos dados coletados.

Assim, ao participar de um experimento, a pessoa recebe instruções iniciais descrevendo o experimento e o protocolo experimental. A seguir deve preencher um formulário de consentimento esclarecido confirmando que está ciente dos objetivos do experimento e está de acordo com a forma que o experimento será conduzido e como os dados coletados serão usados. Esse formulário é assinado pelo experimentador e pelo participante, que também recebe uma cópia para que possa entrar em contato com os experimentadores após o experimento.

2.2.3.1. Desenho experimental

O desenho do experimento define as ações necessárias para responder a pergunta de pesquisa (testar a hipótese). Um passo importante é definir as variáveis do experimento. As variáveis a serem manipuladas são chamadas de variáveis **independentes** enquanto as variáveis a serem medidas são chamadas de **dependentes**. Em IHC, qualquer aspecto mensurável do comportamento humano pode ser uma variável dependente. Há vários tipos de variáveis que podem afetar os resultados, como ilustrado na Figura 2.5 e que são descritas nessa seção.

Voltando ao exemplo de digitação pelo olhar, a velocidade de digitação (medida em número de palavras por minuto [Mackenzie et al. 1999]) constitui a variável dependente, pois depende do que a pessoa participante faz e também do tipo de teclado virtual utilizado para digitação: sem predição de palavras ou AugKey. Da mesma forma, até porque as pessoas participantes não tem como influenciar no tipo de teclado, a variável tipo de teclado (sem predição ou AugKey) corresponde à variável independente desse experimento.

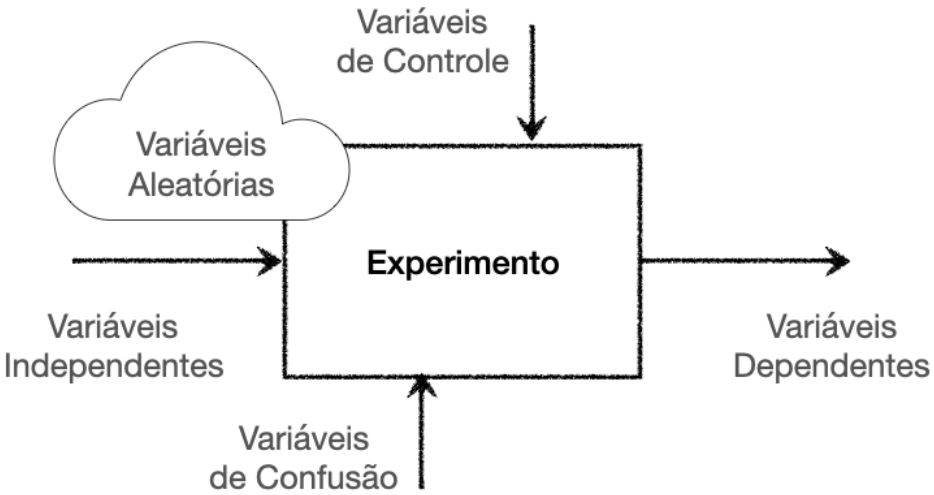

Figura 2.5. Tipos de variáveis a serem consideras no desenho de um experimento.

As variáveis independentes são também chamadas de **fatores** pois é por meio da manipulação desses fatores (mudança de teclado) que medimos os resultados (velocidade de digitação). Experimentos concebidos dessa forma são chamados de **experimentos fatorados** (*factorial experiments*).

Assim como no caso do teclado, a variável independente é tipicamente nominal, com duas (ou mais) categorias. Em nosso exemplo temos dois teclados, o sem predição e o AugKey, mas poderíamos ter outros tipos de teclado também. O número de categorias define os **níveis** (*levels*) da variável independente e são frequentemente chamadas de **condições de teste** (*test conditions*) pois o experimento precisa ser realizado sob cada condição.

Entretanto, as variáveis independentes não precisam estar relacionadas a alguma característica da interface, mas podem estar relacionadas também a características humanas (como sexo, altura, peso, nível de educação, condição econômica, etc) e do ambiente (como nível de barulho, iluminação etc.).

Embora pareça razoável conceber e conduzir um experimento com apenas uma variável independente, é comum conduzir experimentos com múltiplas variáveis imaginando economizar tempo e outros recursos gastos na condução do experimento. No entanto, quanto mais variáveis, mais complexo se torna o experimento e efeitos inesperados entre as variáveis podem comprometer todo o experimento.

Um desenho baseado em apenas um fator possui um **efeito principal** (*main effect*) e nenhum **efeito de interação** (*interaction effect*) entre variáveis. Um desenho com duas variáveis independentes possui dois efeitos principais e um efeito de interação, ou seja, 3 efeitos que devem ser estudados. O efeito de interação é também chamado de **interação por duas vias** (*two-way interaction*) pois é efeito da interação de duas variáveis que não seria sentido na ausência de uma delas. Isso é comum por exemplo em medicamentos que

isoladamente tem efeitos benéficos mas que não devem ser tomados em conjunto pois a interação entre os medicamentos pode ser danosa.

No exemplo de digitação pelo olhar, uma outra variável independente que podemos considerar é o número de sessões que uma pessoa participante deve realizar utilizando cada teclado. Em cada sessão, poderíamos pedir que a pessoa digite um certo número de frases para que possamos avaliar a velocidade de digitação. É comum que após algumas sessões as pessoas aprendam a usar as interfaces e por isso melhorem seu desempenho. Neste caso, podemos avaliar dois efeitos principais (tipo de teclado e número de sessão) e um efeito de interação entre ambas as variáveis independentes.

Imagine agora um experimento com 3 fatores (ou variáveis independentes). No total precisamos considerar 7 efeitos (3 principais, 3 efeitos de duas vias e 1 efeito de 3 vias)! Com quatro fatores, o número de efeitos cresce para 15 (4 principais, 6 de duas vias, 4 de três vias e 1 de quatro vias) e para 5 o número de efeitos a considerar é 31 (5 principais e 26 possíveis efeitos de interação). O elevado número de fatores dificulta a interpretação dos resultados e por isso experimentos com 3 ou mais fatores devem ser evitados.

Como há vários fatores que podem influenciar os resultados mas nem sempre queremos investigar os seus efeitos, esse fatores podem ser controlados e deixados de lado durante o experimento. Esses fatores são chamados de **variáveis de controle**, que podem ser, por exemplo, iluminação do ambiente, configuração do monitor (tamanho, brilho, cor, distância etc.), altura da cadeira, entre outros. A descrição das variáveis de controle e dos valores utilizados permite a reprodução dos experimentos nas mesmas condições.

Entretanto, nem sempre é possível controlar todos os fatores e alguns podem variar aleatoriamente e, por isso, são chamados de **variáveis aleatórias** (*random variables*). Essas variáveis em geral estão associadas a características dos participantes, que inclui sua biometria (altura, peso, força etc.), condição social, nível de estudo etc.

Um perigo que pode ocorrer em um experimento é a existência de **variáveis de confusão** (*confounding variables*), ou fatores de confusão, que correspondem a condições que são alteradas junto com as variáveis independentes. Voltando ao exemplo de avaliação de teclados virtuais, imagine que o tamanho dos teclados usados no experimento são diferentes: a lista de palavras ocupa a tela toda, sendo que as suas teclas são grandes, enquanto o AugKey usa 50% do espaço na tela, portanto suas teclas são menores. Assim, ao variar o tamanho das teclas, a precisão no apontamento por meio do olhar pode ser maior na lista de palavras, provocando menos erros de digitação comparado com o AugKey, que é mais suscetível a erros na estimação do olhar. Esses fatores de confusão nem sempre são tão evidentes quanto nesse exemplo, portanto os pesquisadores precisam ficar atentos a esses fatores para eliminá-los ou, quando não for possível, considerá-los de alguma forma para corrigir os seus efeitos.

2.2.3.2. Escolha da tarefa e participantes

Vamos voltar ao exemplo de desenho de um experimento para a avaliar se o AugKey tem algum efeito sobre a velocidade de digitação em teclados virtuais comparado com um

teclado virtual sem predição de palavras. Digamos que foi decidido realizar o experimento com uma variável independente apenas com dois níveis: sem predição e AugKey.

Que tarefa as pessoas participantes do experimento devem realizar para medir a velocidade de digitação? A escolha da tarefa deve satisfazer dois objetivos:

1. **representar** uma atividade real, comum, natural para cada pessoa realizar usando o teclado; e
2. **discriminar** as condições sendo avaliadas.

Em geral, como em nosso exemplo, a tarefa a ser realizada é evidente, ou seja, como se trata de um teclado virtual, as pessoas participantes devem digitar textos usando cada teclado. Idealmente, os textos deveriam ser os mesmos, mas digitar um texto longo pode causar fadiga e influenciar também nos resultados. Copiar um texto também é inconveniente pois cada participante pode precisar olhar o texto original várias vezes e se perder durante a digitação.

Uma forma de resolver esses problemas é criar uma base de dados com várias frases curtas e neutras, que as pessoas possam memorizar facilmente como ”café com leite, pão e manteiga”. Quando se sentirem prontas, as pessoas iniciam a digitação da frase e, ao terminar, uma nova frase pode ser oferecida. A medida de tempo pode ser automaticamente iniciada com a primeira letra digitada e terminada com o ponto final da frase. Além do tempo, o estímulo (frase) e a resposta (texto digitado) podem ser gravados para posterior processamento e análise.

Dessa forma, espera-se que o único efeito que influencie na velocidade de digitação seja a habilidade dos participantes de usar cada teclado. Mas alguns participantes já podem saber digitar antes do experimento usando um dos teclados mas não o outro. Como eliminar esse efeito?

Podemos selecionar participantes sem nenhuma experiência prévia com nenhum dos teclados, para equilibrar o nível inicial de experiência. Há vários fatores aleatórios no entanto que podem afetar os resultados. Um número suficiente de participantes deve ser utilizado para que os resultados sejam significativos estatisticamente. É difícil determinar quantos mas, se houver trabalhos semelhantes publicados usando dois teclados distintos, que indiquem que resultados significativos foram encontrados com, digamos, 10 participantes, essa pode ser uma boa escolha.

Como vimos, um outro fator importante que deve afetar o resultado é o aprendizado, que faz o desempenho melhorar quanto maior a prática. Se considerarmos que nenhum participante tem experiência, a velocidade de digitação depois de algumas horas de prática deve melhorar, mas o número de horas necessárias para cada participante atingir seu limite de desempenho pode variar bastante.

Estudos longitudinais são experimentos que consideram a quantidade de treinamento como uma variável independente. No caso de pessoas aprendendo a digitar, tipicamente as pessoas ganham bastante velocidade nas primeiras horas até atingir um patamar, onde a velocidade deixa de aumentar. Nesse instante podemos dizer que não há mais efeito do treinamento e podemos avaliar então o efeito do teclado.

Outra decisão importante para o estudo é se devemos separar os participantes em dois grupos distintos, onde cada grupo deve aprender a usar um teclado apenas, ou se todas as pessoas participantes devem aprender a usar os dois teclados.

Quando cada pessoa é testada com cada nível (todas usam os dois teclados), dizemos que a condição do teste é **intra-sujeitos** (*within-subjects*). Essa condição é também chamada de **medidas repetidas** (*repeated measures*) já que cada condição de teste é repetida por cada participante.

Quando cada pessoa é testada com apenas um nível (um dos teclados), dizemos que a condição do teste é **entre-sujeitos** (*between-subjects*). Nesse caso, para o exemplo do teclado, formam-se dois grupos distintos de participantes.

Claramente há vantagens e desvantagens em cada condição. Considerando um mesmo número total de participantes, usando intra-sujeitos teremos mais dados para cada método. No entanto, atingir o patamar nos dois métodos deve requerer um tempo bem maior de treinamento de cada participante. Ao considerar grupos distintos (entre-sujeitos), corremos um risco de um grupo desempenhar melhor devido a fatores aleatórios. No entanto, elimina-se o efeito de interação entre o aprendizado dos dois teclados. Por exemplo, aprender a usar um segundo teclado, após aprender a usar um primeiro, pode ser bem mais rápido e, ao final, obter um desempenho ainda melhor. No caso de estudos intra-sujeitos, uma forma de reduzir esse efeito de interação é alternar as sessões entre o uso de um teclado e outro, de forma que o efeito de aprendizado obtido nas sessões anteriores seja distribuído.

Por exemplo, em um estudo longitudinal que mede a velocidade de digitação diariamente, uma pessoa poderia fazer duas sessões de testes por dia, a primeira usando sem predição e a segunda usando AugKey. No dia seguinte, a ordem é invertida. Na primeira sessão, metade dos participantes devem começar com um teclado e a outra metade com o outro. Um outro desenho possível seria fazer metade do grupo começar a treinar com um método até atingir o patamar e só aprender a utilizar o segundo teclado após aprender o primeiro. Como cada metade começa aprendendo um teclado diferente, pode ser possível também inferir se há algum efeito na velocidade de aprendizado de um segundo teclado.

Para experimentos com duas variáveis independentes, é possível também utilizar um desenho misto, onde uma variável é testada intra-sujeitos e a outra entre-sujeitos. No caso dos teclados, podemos considerar se há diferença na curva de aprendizado entre homens e mulheres. Nesse caso, considerando o fator treino com por exemplo 10 níveis (10 sessões de treino com um teclado) e o fator sexo com dois níveis (feminino e masculino), o fator treino pode ser testado intra-sujeitos e o fator sexo de forma entre-sujeitos.

2.2.4. Coleta dos dados

Após definido todo o desenho experimental e preparado os materiais (hardware e software), deve-se conduzir um experimento piloto com pelo menos 2 ou 3 voluntários para testar se todo o processo está satisfatório e nenhum detalhe importante foi esquecido. Esses voluntários podem fazer parte do grupo de pesquisadores que, com uma atitude neutra, devem avaliar se todas as etapas do experimento, desde a recepção dos participantes, sua instrução, esclarecimento e assinatura do termo de consentimento esclarecido,

treinamento da tarefa, execução do experimento, até a despedida, foram implementadas satisfatoriamente.

O piloto é importante para revelar pequenos problemas ou inconveniências que devem ser resolvidas antes de iniciar a coleta verdadeira, com o protocolo experimental revisado e corrigido.

Durante o experimento, é necessário também que os experimentadores se conduzam de forma neutra para não influenciar, positiva ou negativamente, os participantes. Por exemplo, frases como ”agora você vai usar esse teclado que é bem melhor que o primeiro”, devem ser evitadas.

2.2.5. Avaliação dos resultados

Lembre-se que o objetivo do experimento é responder uma pergunta de pesquisa, comprovando ou refutando uma hipótese. Para o exemplo dos teclados, o experimento deve coletar dados de várias pessoas e a seguir precisamos comprovar se há diferença entre os desempenhos medidos de cada teclado. Chamamos de **hipótese nula** o caso em que não há diferença, ou seja, todos os teclados apresentam o mesmo desempenho. Os testes de hipótese são métodos estatísticos que podem comprovar ou refutar a hipótese nula.

O resultado de um teste é denominado estatisticamente significativo se for considerado improvável de ter ocorrido por acaso, assumindo a hipótese nula verdadeira. Para isso, o teste calcula uma probabilidade (valor p) e caso p seja menor que um limite pré-especificado (nível de significância, tipicamente 5%), a hipótese nula pode ser rejeitada. Caso contrário, dizemos que não há diferença significativa. No caso dos teclados, isso serviria como evidência que o AugKey não tem efeito (nem melhora nem piora) sobre a velocidade de digitação usando um teclado sem predição de palavras.

A escolha do método a ser empregado depende do tipo de dado coletado. Testes **não paramétricos** são tipicamente usados para o tratamento de dados nas escalas ordinal e nominal. Dados intervalares e proporcionais são tipicamente avaliados por meio de testes **paramétricos**. Como esses dados podem ser reduzidos a dados nominais ou ordinais, eles também podem ser tratados por testes não paramétricos.

Testes paramétricos são assim chamados pois dependem de distribuições de probabilidade, como uma distribuição normal ou distribuição t, enquanto os testes não paramétricos não assumem qualquer distribuição em particular.

Um procedimento padrão para analisar dados nominais (em escala nominal ou categórica) é o teste chi-quadrado (χ^2). Nesse teste, os dados são organizados na forma de uma tabela de contingência, onde cada linha e coluna contém as frequências das observações de cada categoria. O teste χ^2 compara as frequências observadas com os valores esperados, assumindo que todas as categorias devem possuir comportamentos similares (hipótese nula). A aplicação do teste confirma ou não essa hipótese.

Para dados ordinais os testes mais comuns são o teste de Mann-Whitney, o teste de postos sinalizados de Wilcoxon, o teste de Kruskal-Wallis e o teste de Friedman [Robertson and Kaptein 2018]. Esses testes seguem o mesmo procedimento geral de calcular uma grandeza estatística relativa ao teste usando os dados para rejeitar ou não a hipótese nula.

Apesar da relevância dos testes não paramétricos, os testes paramétricos são os mais utilizados em pesquisas na área de IHC por serem capazes de tratar os dados quantitativos resultantes de medidas de desempenho humano. A próxima seção introduz a análise de variância ou ANOVA, por se tratar do teste paramétrico mais comum e amplamente utilizado.

2.3. Análise de variância

A ANOVA, ou teste-F, é uma forma de teste de hipótese estatística amplamente utilizada na análise de dados em experimentos fatorados. ANOVA pode ser usada tanto em estudos entre-sujeitos quanto em estudos intra-sujeitos. Também é aplicável quando existe mais de uma variável independente.

Vamos supor que nosso estudo tem apenas uma variável independente, como no exemplo dos teclados virtuais, onde a variável independente é o tipo de teclado e possui dois níveis: sem predição e AugKey. Como em todo teste estatístico, na ANOVA a hipótese nula é que não existe diferença significativa entre as médias de cada condição experimental: as médias de velocidade de digitação com o teclado sem predição e com o teclado com AugKey são similares. Esta é a hipótese que o teste ANOVA irá rejeitar ou não, dado o nível de significância preestabelecido.

Quando a hipótese nula é rejeitada (pois obtivemos um valor de p menor que o nível de significância, usualmente $p < 0.05$), podemos concluir que a variável independente tem um efeito significativo na variável dependente. Ou seja, existem pelo menos dois níveis da variável independente cujas médias são significativamente diferentes. Como temos apenas 2 níveis (tipos de teclado), então podemos concluir que a diferença entre as duas é significativa. Este é o caso mais simples de aplicação da ANOVA. Em caso de existir diferença significativa, uma simples conferência da média, nos revela qual método apresentou melhor desempenho nos experimentos.

O que acontece quando existem mais de 2 níveis, por exemplo, se estivermos comparando 3 ou mais tipos de teclado? A interpretação é a mesma que para 2 níveis: a variável independente tem um efeito significativo na variável dependente, porém ANOVA não nos diz exatamente quais níveis são diferentes entre si.

Para entender melhor, vamos supor que nossa variável independente tem três níveis: A, B e C. Um resultado significativo de ANOVA significa que há uma diferença significativa entre as médias de alguns dos grupos, mas que pode ser entre A e B, B e C, A e C ou mais de uma combinação ao mesmo tempo. Para saber onde, precisamos fazer um teste conhecido como *post-hoc*.

Os testes *post-hoc* complementam um resultado significativo de ANOVA, nos ajudando a descobrir entre quais níveis a diferença é significativa. O mais comum é comparar todos os níveis da variável independente dois a dois. Se temos três condições A, B e C, temos no total 3 comparações: A $\times$ B, A $\times$ C e B $\times$ C. Não entraremos em detalhes sobre como os testes *post-hoc* são calculados, mas os mais comuns são o teste de Student com correção Bonferroni ou Holm e o teste de Scheffe [Robertson and Kaptein 2018].

participante	p0	p1	p2	p3	p4	p5	p6	p7	p8	p9
Sem Predição	5.3	3.5	5.1	3.6	4.6	4.1	4.0	4.8	5.2	5.1
Augkey	5.7	4.8	5.1	4.6	6.1	6.8	6.0	4.6	5.5	5.6

Tabela 2.1. Tempos médios obtidos por participante no Exp1.

participante	p0	p1	p2	p3	p4	p5	p6	p7	p8	p9
Sem Predição	2.4	2.7	3.3	6.1	6.7	5.4	7.9	1.2	3.0	6.6
Augkey	6.9	7.3	2.9	1.8	7.8	9.2	4.4	6.6	4.8	3.1

Tabela 2.2. Tempos médios obtidos por participante no Exp2.

2.3.1. Exemplo de ANOVA

Agora vamos aprofundar um pouco mais sobre o teste ANOVA. Para facilitar o entendimento dessa técnica, vamos considerar dois experimentos hipotéticos distintos, Exp1 e Exp2, ambos com o mesmo objetivo e metodologia para avaliar o desempenho de pessoas usando os teclados sem predição de palavras e o AugKey. Considere que 10 pessoas participaram de cada experimento e que a velocidade média de digitação medida em caracteres por minuto foi calculada para cada participante usando algumas frases neutras. A média dos tempos obtidos por participante em todas as sessões para o Exp1 são mostradas na Tabela 2.1 e as médias para o Exp2 são mostradas na Tabela 2.2.

Uma vez tabelados os dados, a média das médias pode ser calculada considerando todos os participantes como mostrado na Figura 2.6, indicando o desempenho de cada teclado. Observe que as velocidades médias usando cada teclado são as mesmas nos dois experimentos. Apesar do gráfico mostrar uma vantagem do AugKey, apenas no Exp1 observou-se diferença significativa entre as velocidades.

Na aplicação de ANOVA nesse exemplo de teclados, a hipótese nula seria que todas as pessoas participantes possuem o mesmo desempenho usando os dois modelos, ou seja, não há efeito do tipo de teclado (sem predição ou AugKey) sobre a velocidade de digitação. No entanto, como a velocidade atingida por cada participante deve ser diferente, podemos considerar cada velocidade medida como uma amostra aleatória da mesma população. A rejeição da hipótese nula significa que as diferenças nas velocidades entre os grupos de participantes provavelmente (dentro do nível de significância) não são devidas ao acaso.

Os testes-F recebem esse nome em homenagem ao estatístico Ronald Fisher. A estatística-F é simplesmente uma razão de duas variâncias e pode ser definida como:

$$F = \frac{\text{variância entre as médias das amostras}}{\text{variância dentro das amostras}}$$

Para entender essa fórmula, vamos voltar ao experimento onde 10 participantes usaram os teclados sem predição de palavras e AugKey. Porque a diferença foi significativa no gráfico à esquerda mas não foi significa no da direita? Você já deve ter percebido pelas tabelas que a resposta está na variância das amostras, ou seja, os valores obtidos no experimento Exp2 estão mais espalhados (distantes da média) que no Exp1.

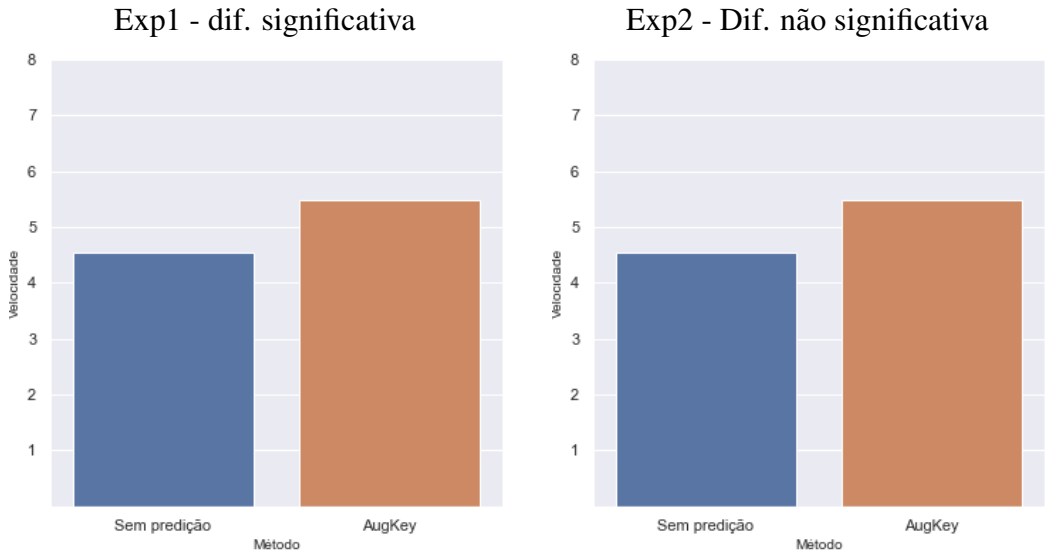

Figura 2.6. Médias obtidas nos experimentos Exp1 e Exp2. Ambos os experimentos avaliam a velocidade de digitação usando os teclados sem predição de palavras e AugKey. Observe que as médias nos dois experimentos é a mesma. Mas observe também que apenas no Exp1 houve diferença significativa nos resultados.

A variância é calculada como o quadrado do desvio padrão e indica a dispersão dos dados em relação à média. Variâncias pequenas indicam que os dados não fogem muito da média. A vantagem de usar o desvio padrão é que sua unidade é a mesma da média e portanto mais fácil de entender (pode ser colocada no gráfico da média por possuir a mesma unidade).

Como podemos observar na Figura 2.7, o desvio padrão, representado pelas linhas verticais no centro das barras que indicam as médias, é menor no Exp1 e maior no Exp2 (gráfico da direita). Com isso podemos afirmar que o valor de F no Exp1 será maior que o valor de F no Exp2, pois a variância dentro das amostras é maior no Exp2.

A Tabela 2.3 mostra os dados de média, desvio padrão, variância e resultado do teste ANOVA para esse exemplo. Como podemos observar, o valor de F foi maior no Exp1 e o valor de p foi menor que 0.05, indicando que a diferença nas médias de velocidade de digitação nos teclados foi estatisticamente significativa. Já no Exp2 o valor de F foi menor e o valor de p acima de 0.05, indicando que a diferença não foi estatisticamente significativa.

Tabela 2.3. Dados de média, desvio padrão, variância, valor de F e valor de p para os dados de Exp1 e Exp2.

Métrica	**Experimento 1**		**Experimento 2**	
	Sem predição	AugKey	Sem predição	AugKey
Média	4.53	5.48	4.53	5.48
Desvio padrão	0.65	0.68	2.15	2.31
Variância	0.42	0.46	4.64	5.33
F	10.5		0.7	
p	**0.01**		0.4	

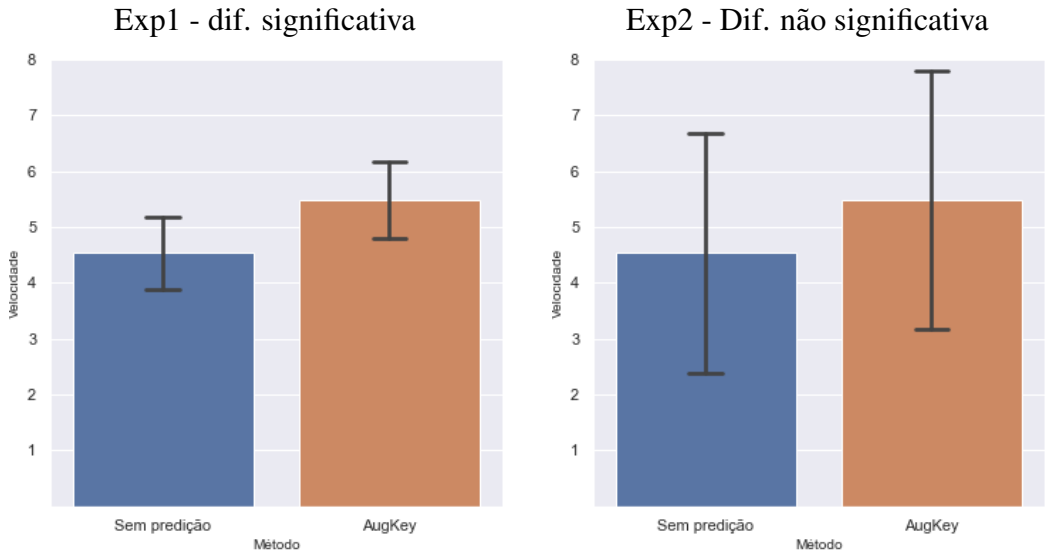

Figura 2.7. Médias e desvios padrão obtidos nos experimentos Exp1 e Exp2. Observe que o desvio é bem maior em Exp2 e por isso os resultados de Exp2 não apresentam diferença significativa.

A modo de exemplo, vamos calcular o valor de F para o Exp1 A variância total observada nos dados coletados em um experimento intra-sujeitos pode ser quebrada em três termos: a variância explicada pelas condições, a variância explicada pelas variações entre os participantes e a variância que não pode ser explicada (o erro residual).

Para calcular o valor de F, primeiro vamos calcular o valores descritos a seguir:

- Soma dos quadrados total: SS_t é a variância total, considerando todas as amostras juntas. No Exp1 $SS_t = 13.3$;
- Soma dos quadrados entre as condições: SS_b é a parte da variância justificada pelas diferentes condições experimentais (por exemplo, por usar um teclado sem predição e outro com AugKey). No Exp1 $SS_b = 4.51$;
- Soma dos quadrados dos participantes: SS_s é a variância explicada pelas características de cada participante no desempenho das tarefas, que no Exp1 é $SS_s = 3.87$;
- Soma dos quadrados do erro: SS_e é a variância que não é explicada nem pelas condições experimentais e nem pelos participantes (erro residual). Este pode ser calculado a partir dos outros valores. Como já sabemos, a soma dos quadrados total é a soma dos três já estudados:

$$SS_t = SS_b + SS_s + SS_e.$$

Logo temos que

$$SS_e = SS_t - SS_b - SS_s = 3.87.$$

Até agora calculamos a soma dos quadrados, mas precisamos calcular a média de cada soma. Para isso precisamos apenas dividir pelos graus de liberdade de cada um. O **grau de liberdade** da soma dos quadrados das condições é igual ao número de condições menos um, já o grau de liberdade da soma dos quadrados do erro é igual ao número de condições menos um vezes o número de participantes menos um. Vamos ver um exemplo.

As médias de interesse para calcular F são calculadas assim:

$$MS_b = \frac{SS_b}{\text{n. condições - 1}} = \frac{4.51}{2-1} = 4.51$$

e

$$MS_e = \frac{SS_e}{\text{(n. condições - 1)} * \text{(n. part. - 1)}} = \frac{3.87}{(2-1)*(10-1)} \frac{3.87}{9} = 0.43$$

Finalmente, o valor de $F = \frac{MSb}{MSe} = 10.5$. Os graus de liberdade de F são (n. condições - 1, (n. condições - 1)*(n. part. - 1)) = (1, 9). Logo podemos afirmar que F(1, 9) = 10.5 .

Agora, como descobrir se esse valor de F representa uma diferença significativa entre os dois teclados? Para isso é necessário consultar uma tabela de valores críticos de F. Esta tabela tem por colunas o grau de liberdade do numerador na expressão de F e por linhas os graus de liberdade do denominador. Após localizar o valor correspondente aos graus de liberdade da expressão de F, comparamos o valor da tabela com o F calculado: se o valor de F for maior ou igual ao valor da tabela, então o resultado foi estatisticamente significativo. Caso o valor e F for inferior ao valor da tabela, dizemos que o resultado não é significativo, ou seja, que não houve um **efeito** significativo. É importante levar em consideração que cada tabela serve para um determinado nível de significância, no nosso caso usamos o valor de 0.95 como nível de significância, mas caso outro nível seja usado (por exemplo 0.99 ou 0.90), será necessário consultar a tabela de valores de F correspondente a esse nível. Na prática porém, o mais comum é usar um nível de significância de 0.95.

Como vimos, a tabela de valores de F não nos fornece o valor exato de p, mas nos diz se o resultado é significativo ou não para um determinado nível de significância. Na prática, a maioria das bibliotecas estatísticas já calculam o valor exato de p quando executam o teste ANOVA e assim, podemos comparar o valor de p com (1 - nível de significância) e, se o valor de p for menor, então o resultado é estatisticamente significativo. No exemplo Exp1, o valor de p calculado pelo software resultou em 0.01. Como 0.01 ¡ 0.05, o resultado é estatisticamente significativo. Note que o valor 0.05 é o resultado de 1 - 0.95, que é o nível de significância do nosso estudo. Uma outra forma de pensar nesses resultados é que a chance (probabilidade) de que os resultados serem diferentes é menor que 5%.

Como podemos reportar o resultado do teste ANOVA? Um outro fator importante da pesquisa é que seus resultados devem ser publicados. Afinal, encontrar uma diferença significativa é um resultado muito positivo para qualquer pesquisa. Recomendamos seguir o padrão sugerido na 6^a Edição do Manual da Associação Americana de Psicologia (*APA*, do inglês *American Psychological Association*), disponível em http://www.apastyle.com.

Esse padrão sugere que todos os valores estatísticos sejam arredondados para duas casas decimais, com exceção dos valores onde o nível de significância (p) seja menor que 0.001, em cujo caso deve ser usada a notação "$p < 0.001$".

No caso de uma diferença estatisticamente significativa, como no Exp1, podemos escrever assim: *um teste ANOVA de 1 via mostrou um efeito significativo do tipo de teclado na velocidade da digitação, F(1, 9) = 10.5, p = 0.01*.

2.4. Processamento e análise de dados com Python

As medidas quantitativas coletadas durante os experimentos são tipicamente armazenadas na forma de tabelas que podem ser processadas por planilhas eletrônicas tipo Excel. No entanto, a exploração dos dados e a visualização de resultados pode ser feita de forma mais eficiente usando outros softwares voltados para análise de dados como Matlab, Octave, IDL e SciLab.

Nesse curso vamos utilizar o JupyterLab (https://jupyter.org), um ambiente web interativo que permite desenvolver Jupyter Notebooks em Python (e que pode ser também utilizado com outras linguagens como R e Julia). Um Jupyter Notebook é um documento que pode misturar texto, código, equações e visualizações que facilitam a exploração dos dados.

Enquanto R é uma linguagem dedicada ao processamento estatístico dos dados e portanto adequada à análise experimental, adotamos Python nesse curso por ser uma linguagem de programação de propósito geral, poderosa, flexível e mais fácil de aprender por possuir uma sintaxe simples. Isso deve facilitar o acompanhado dessa seção mesmo pelas pessoas com pouca experiência de programação e/ou experiência com outras linguagens. Por ser uma linguagem interpretada, ela também permite o desenvolvimento rápido de aplicativos em muitas áreas distintas, incluindo a análise exploratória de dados. Todos os exemplos que ilustram esse curso foram desenvolvidos usando Jupyter Notebooks em Python e estão disponíveis no endereço https://ime.usp.br/~hitoshi/jai2021.

2.4.1. Um pouco de Python

Nesse curso, vamos assumir que você já possui alguma experiência com uma linguagem de programação imperativa, como C, C++, Java ou Python (suficiente, por exemplo, para manipular uma estrutura de dados indexada como uma lista, vetor ou matriz) e também algum conhecimento de estatística (como saber calcular a média e a variância de um conjunto de valores). Assim podemos focar no uso do Python para a exploração e análise dos dados.

Caso você não tenha muita experiência e esteja interessado em um curso introdutório de programação em Python, recomendamos o material que utilizamos em nosso curso de introdução à computação disponível no endereço https://panda.ime.usp.br/pensamentos. Esse material é utilizado também no curso Introdução à Ciência da Computação com Python no Coursera, ministrado pelo Prof. Fábio Kon. Outra referência que recomendamos caso você deseje aprender probabilidade e estatística com Python é o livro *Think Stats* do Prof. Allen Downey disponível no endereço (https://greenteapress.com/wp/think-stats-2e), embora essa última referência esteja disponível

apenas em inglês.

O Python (https://www.python.org) oferece uma extensa coleção de módulos disponíveis gratuitamente para as principais plataformas computacionais modernas, como Linux, MacOS e Windows. Nessa seção vamos visitar alguns módulos que lhe podem ser úteis no processamento e análise estatística dos dados coletados em seus experimentos.

Caso você deseje instalar Python em seu computador para acompanhar os exemplos desse curso, sugerimos a instalação do pacote Anaconda (http://anaconda.com), desenvolvido para usar Python na área de ciência de dados. O Anaconda facilita a instalação e manutenção de todos os módulos e ferramentas que vamos utilizar nessa seção, como o JupyterLab, Numpy, Seaborn, Pandas e SciPy. Você pode baixar e utilizar a versão individual do Anaconda gratuitamente.

2.4.1.1. Criando e visualizando dados usando Numpy e Seaborn

Vamos iniciar com a descrição dos módulos Numpy e Seaborn como ilustrado no trecho de programa 2.1.

A primeira linha do programa 2.1 carrega o módulo Numpy (Numerical Python) sob o nome `np`. O Numpy inclui funções eficientes para criação e manipulação de matrizes (qualquer estrutura n-dimensional), que inclui operações matemáticas, lógicas, álgebra linear, estatística, transformações etc. O Numpy se tornou em um módulo fundamental para a computação científica. Vários outros módulos científicos e matemáticos são derivados do Numpy, alguns dos quais veremos nessa mesma seção. Uma estrutura de dados fundamental do Python é a lista (tipo `list`), que permite manipular uma sequência indexada com elementos de qualquer tipo, inclusive outras listas. Uma lista em Python utiliza colchetes (”[]”) como delimitadores. Por exemplo:

```
lista = [ 12, 3.14, 'hello', [5, 6]],
```

corresponde a uma lista com 4 elementos, o inteiro `12` na posição 0, o float `3.14` na posição 1, a string `'hello'` na posição 2, e a lista `[5, 6]` na posição 3 da lista.

Programa 2.1. Uso de Numpy e Seaborn

```
1  import numpy as np
2  import seaborn as sns
3
4  pts  = np.linspace(-np.pi , np.pi, 30)
5  valCos = np.cos(pts)
6  graf = sns.scatterplot( x=pts, y=valCos )
```

A estrutura básica do Numpy é o **ndarray** (vetor n-dimensional), que permite representar elementos de um mesmo tipo (ou seja, ao contrário de listas, seu conteúdo é homogêneo). Enquanto as listas são alocadas dinamicamente, o tamanho de um ndarray não é mais alterado após sua criação. A alteração de tamanho implica na criação de um novo ndarray. Uma lista homogênea pode ser convertida para um ndarray usando a função de conversão `array()` como

```
nda = np.array([1, 2, 3, 4]).
```

Apesar (ou devido) a essas limitações, as operações com Numpy são executadas de forma mais eficiente e com menos código do que usando listas pois o Numpy permite operações vetoriais, evitando o uso explícito de laços.

Um exemplo de operação vetorial pode ser visto na linha 5. Na linha 4, a função `np.linspace()` do Numpy (observe a notação com ponto que indica módulo.função) cria um ndarray com 30 pontos no intervalo [-π, π], uniformemente espaçados, e os atribui à variável `pts`. A função np.cos() é aplicada a todos os pontos do ndarray, sem necessidade de criar um laço explícito para percorrê-lo. Os valores resultantes são atribuídos à variável `val_cos`.

A segunda linha do programa 2.1 carrega o módulo Seaborn sob o nome `sns`. O Seaborn (https://seaborn.pydata.org/) é baseado no Matplotlib, um módulo do Python para criação de visualizações de dados bastante flexível e utilizado, por exemplo, para a apresentação de gráficos. O Seaborn é um módulo mais simples que o Matplotlib, voltado a criar gráficos visualmente mais atraentes. Mas caso necessário, o Seaborn pode ser utilizado em conjunto com o Matplotlib para combinar seus recursos.

A linha 6 mostra como é simples criar um gráfico de pontos (*scatterplot*) usando Seaborn. Nesse caso, basta associar o vetor de pontos horizontais em `pts` com seus valores correspondentes no eixo vertical em `valCos`. Observe que o gráfico é atribuído à variável `graf`, para podermos alterar certas propriedades mais tarde, como os nomes de cada eixo.

2.4.2. Explorando dados usando Jupyter Lab

O Jupyter Lab (https://jupyterlab.readthedocs.io) é um ambiente web (ou seja, roda em um nagegador da Internet como o Mozzila Firefox, Google Chrome ou Microsoft Edge) que fornece blocos de construção flexíveis para computação exploratória interativa. Embora o Jupyter Lab tenha muitos recursos encontrados em ambientes de desenvolvimento integrado (conhecidos como IDEs do inglês *Integrated Development Environment*) tradicionais, ele permanece focado na computação exploratória e interativa.

A interface do Jupyter Lab pode ser vista na Figura 2.8, rodando dentro do navegador Mozilla Firefox. A figura mostra o navegador de arquivos (*file browser*) aberto, que mostra os Jupyter Notebooks (arquivos com a extensão ".ipynb") disponíveis na pasta do projeto. Quando esses arquivos são abertos, como ilustrado na figura, cada arquivo recebe uma aba na área principal de trabalho (para edição/exploração de dados), como se fossem páginas distintas do navegador.

A Figura 2.8 mostra o conteúdo do Jupyter Notebook armazenado no arquivo "exemplo01.ipynb", que corresponde ao programa 2.1, após a execução do bloco. A área de trabalho atua como um ambiente de desenvolvimento Python, permitindo a edição, visualização e testes do programa. Cada bloco pode ser executado de forma independente dos demais e o resultado é exibido instantaneamente, logo abaixo do bloco executado. No entanto, como todos os blocos compartilham do mesmo interpretador (iPython kernel), a ordem de execução dos blocos pode afetar o resultado. Essa flexibilidade facilita a exploração de dados mas pode se tornar confusa em problemas mais complexos. Por isso

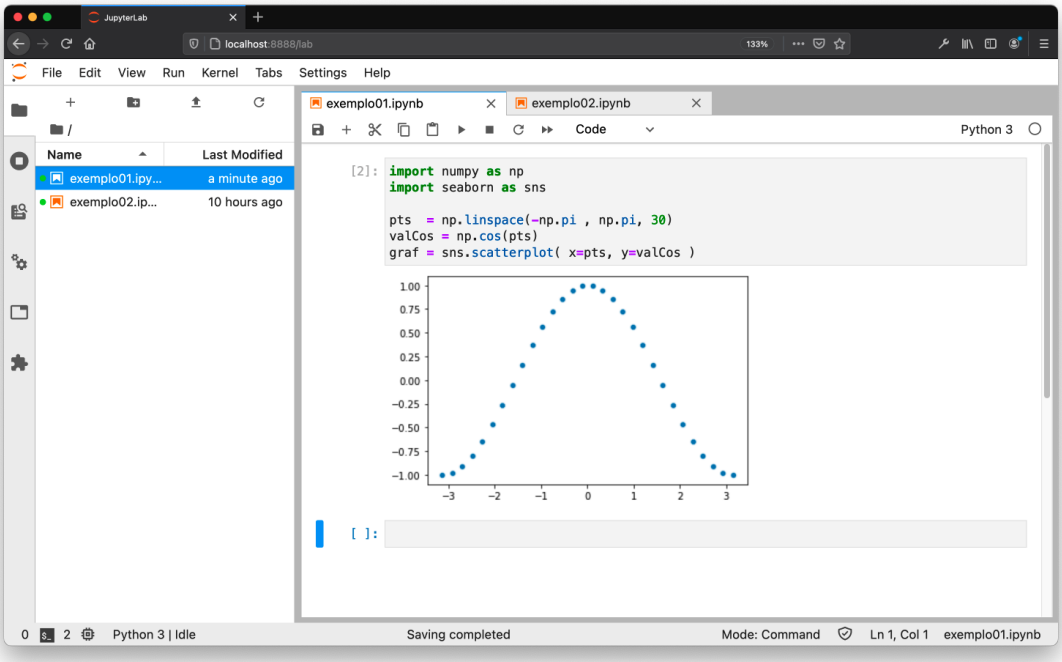

Figura 2.8. Interface do Jupyter Lab

o iPython kernel pode ser reinicializado quando necessário. O Jupyter Notebook permite também alterar a ordem dos blocos e selecionar apenas um grupo de blocos para serem executados.

Para ilustrar essa característica, a Figura 2.9 mostra a mesma interface agora com o navegador de arquivos fechado e, após o gráfico de pontos, vemos que a pessoa usuária digitou um comando Python para imprimir a média e a variância de `valCos`. Ao escrever a linha com a função `print()`, executamos a linha algumas vezes, por exemplo para alterar o padrão como limitar em 3 o número de dígitos após a vírgula, e por isso executamos o mesmo bloco algumas vezes. Por isso o número entre colchetes desse bloco aparece como ”[5]” na figura.

2.4.3. Pandas: processamento de dados em tabelas

O Numpy oferece várias funções para a geração de dados sintéticos segundo algumas distribuições comuns que são muito úteis para simulação e testes. Vamos criar uma tabela com dados sintéticos para introduzir o Pandas e usar um pouco mais do Jupyter Notebook.

O Pandas (https://pandas.pydata.org) é um módulo do Python voltado para a análise de dados usando **DataFrames**, uma estrutura equivalente a uma tabela ou planilha (aliás, a origem do nome ”Pandas” vêm da combinação das palavras ”pannel” + ”data” em inglês). Embora o Pandas ofereça muitos recursos próprios, nesse curso vamos utilizar o Pandas apenas para carregar, salvar e manipular tabelas. Um DataFrame é diferente de um ndarray de duas dimensões do Numpy pois cada coluna possui nomes, e os tipos de dados de cada coluna podem ser distintos, além de fornecer mecanismos elaborados para seleção de dados e sua manipulação.

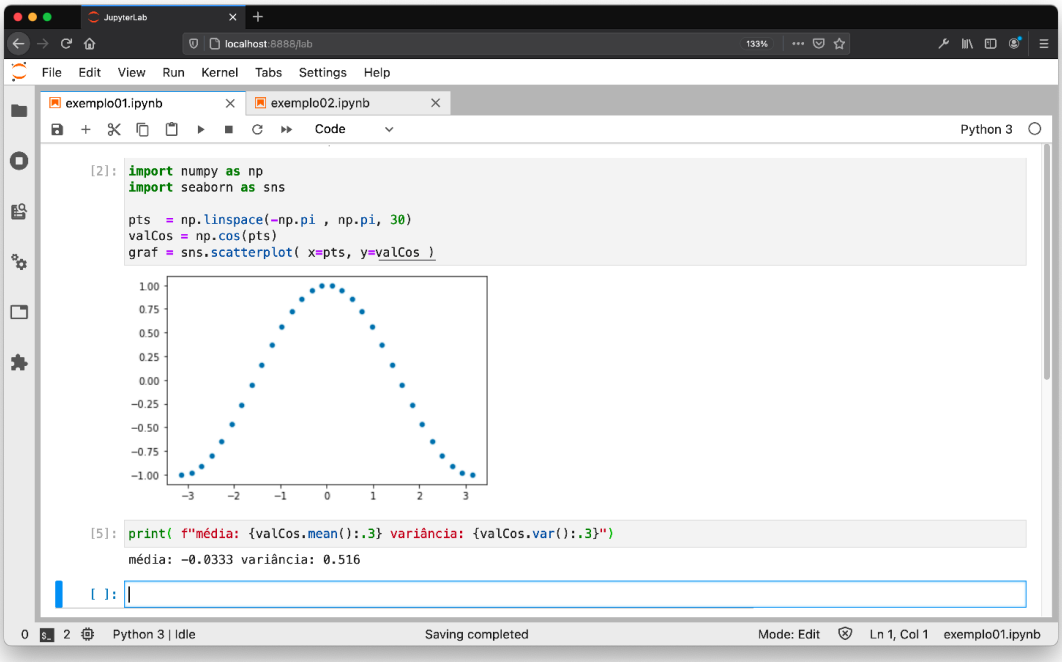

Figura 2.9. Interface do Jupyter Lab: cada bloco (indicado pelo número em colchetes) contém um pedaço de código que pode ser explorado de forma independente dos demais.

Programa 2.2. Criação de um DataFrame

```
1  import pandas as pd
2
3  valSin = np.sin(pts)
4  dicio = {'xcoords':pts, 'seno':valSin, 'cosseno':valCos}
5  dframe = pd.DataFrame( dicio )
6
7  dframe.to_csv( 'senocos.csv', sep=';' )
```

O trecho de código 2.2 ilustra como criar um DataFrame. Você pode colocar esse código em um novo bloco do Jupyter Notebook e executá-lo. A linha 1 mostra que o módulo `pandas` é carregado com o nome `pd`. Estamos considerando que o trecho de código 2.1 já tenha sido executado e portanto a variável `pts` já esteja carregada e pode ser usada para criar um novo ndarray com os valores do seno de `pts`.

Um DataFrame pode ser criado a partir de um dicionário do Python como nas linhas 4 e 5. Na linha 4, as chaves do dicionário indicam o nome de cada coluna, e os valores da coluna são definidos como uma lista ou ndarray. A linha 5 cria o DataFrame a partir do dicionário `dicio` e na linha 7 o DataFrame `dframe` é salvo no arquivo de nome ”senocos.csv” e o caractere ’;’ como separador de colunas.

A Figura 2.10 ilustra o estado do Notebook após a criação e execução do bloco 2.2. A figura mostra o navegador de arquivos aberto, onde podemos notar que o arquivo ”senocos.csv” foi criado no mesmo diretório do projeto.

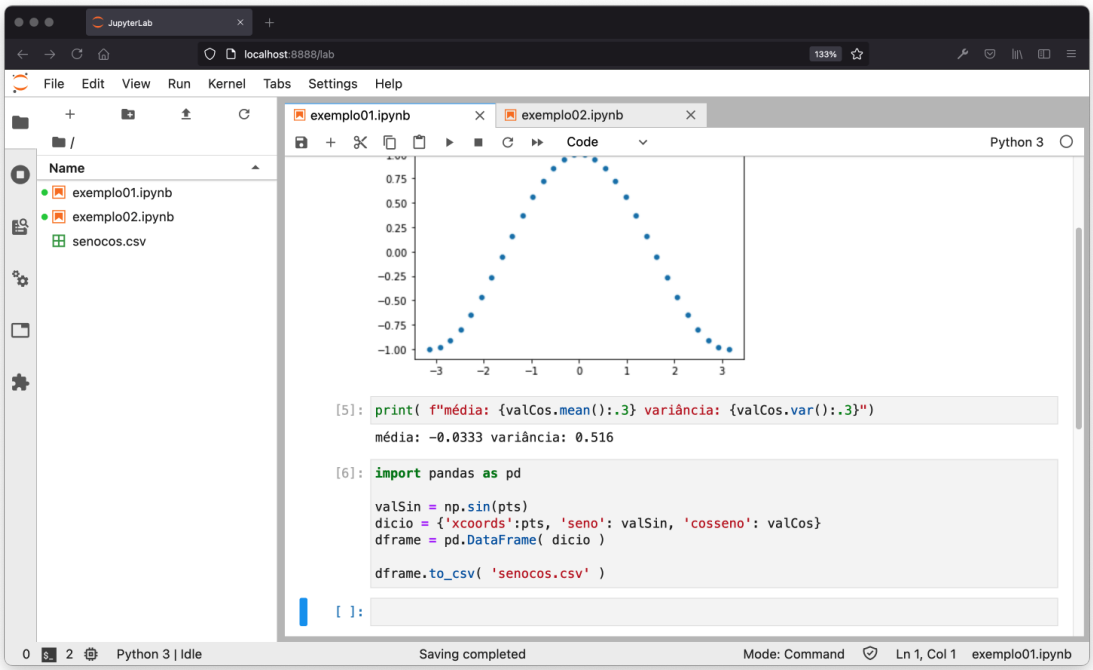

Figura 2.10. Notebook após a execução de um novo bloco para criação de um DataFrame.

Assim como o ”senocos.csv”, os resultados dos experimentos são tipicamente representados na forma de tabelas e armazenados em arquivos semelhantes. Há vários outros formatos possíveis, como o formato XLS ou XLSX usados pelo Excel. Nesse curso vamos adotar o formato CSV (*Comma Separated Values*) por ser simples e facilmente portável para outras aplicações, inclusive para o Excel. Outra vantagem desse formato é que seu conteúdo é legível e o caractere separador entre colunas pode ser redefinido, ou seja, não precisa ser uma ”vírgula” e podemos adotar outros caracteres como ”ponto-e-vírgular” (;) ou ”dois-pontos” (:).

O trecho abaixo mostra as primeiras linhas do arquivo ”senocos.csv” salvo pelo Pandas, onde as colunas aparecem separadas pelo caractere ”;”. Observe que a primeira linha contém os nomes das colunas (que podem corresponder ao nome das propriedades medidas em um experimento) e cada linha seguinte contém uma amostra com os valores das medidas realizadas de cada propriedade. Observe que o Pandas inicializa a primeira coluna com o índice de cada amostra. Como essa coluna não recebe um nome, a primeira linha do arquivo já começa com um ponto-e-vírgula.

```
;xcoords;seno;cosseno
0;-3.141592653589793;-1.2246467991473532e-16;-1.0
1;-2.9249310912732556;-0.21497044021102427;-0.9766205557100867
2;-2.708269528956718;-0.4198891015602648;-0.9075754196709569
...
```

Para testar se o arquivo foi salvo corretamente, experimente digitar e executar o trecho de código 2.3 em outro bloco do seu Jupyter Notebook.

Programa 2.3. Leitura de um DataFrame

```
1    salvo = pd.read_csv( 'senocos.csv' )
2    print( salvo )
```

2.4.4. Visualizando múltiplos dados

Como vimos o Seaborn permite criar visualizações simples rapidamente e com gráficos visualmente agradáveis. Mas para criar gráficos mais elaborados, precisamos recorrer ao módulo **Matplotlib**. Por exemplo, para plotar os dados de seno e cosseno no mesmo gráfico vamos usar o pacote **Pyplot** que faz parte do Matplotlib.

Programa 2.4. Visualizando múltiplos dados

```
1
2    from matplotlib import pyplot as plt
3    sns.set()
4
5    fig, ax1 = plt.subplots()
6    ax2 = ax1.twinx()
7    ax1.plot(dframe['xcoords'], dframe['cosseno'])
8    ax2.scatter(dframe['xcoords'], dframe['seno'], color='r')
9
10   plt.show()
11   fig.savefig('sencos.png')
```

O trecho de código 2.4 ilustra uma forma para combinar os dados do seno e do cosseno usando Pyplot. Digite esse programa em um novo bloco e verifique o resultado, que é mostrado na Figura 2.11. Esse trecho começa carregando o `pyplot` com o nome `plt`. Apesar de usar recursos do Pyplot, o comando na linha 2 especifica que os gráficos gerados devem continuar usando o padrão visual do Seaborn.

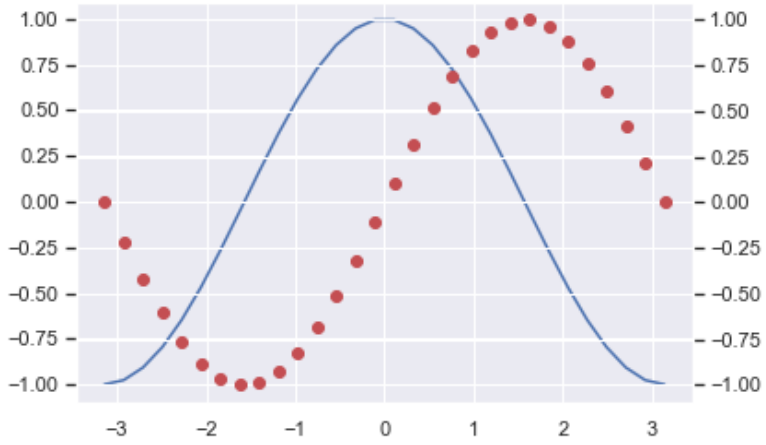

Figura 2.11. Gráfico gerado pelo trecho de código 2.4.

A função `subplots()` chamada na linha 5 devolve dois objetos: a figura e

um eixo (*axis*) para plotar o conteúdo gráfico. No caso das funções seno e cosseno, criamos um segundo eixo `ax2` na linha 6 para plotar um segundo conjunto de dados, mas compartilhando o mesmo eixo horizontal. Na linha 7, o conjunto `cosseno` é plotado na forma de uma **linha contínua** (usando `plot()`) no eixo `ax1`, usando os valores na coluna `'xcoords'` do DataFrame como pontos no eixo x. Na linha 8 o conjunto `seno` é plotado na forma de **pontos** (usando `scatter()` no eixo `ax2`, na cor vermelha (`('r'ed)` – experimente `'b'lue`, `'g'reen`, etc).

Há várias outras opções que podem ser configurados (consulte a documentação do Seaborn e Pyplot) e, quando terminar de configurar, a linha 10 mostra como fazer o Pyplot exibir o gráfico. A figura pode também ser salva usando o método `savefig` como mostrado na linha 11.

Programa 2.5. Visualizando múltiplos dados separadamente

```
1    fig2, eixos = plt.subplots(1,2, figsize=(10,5))
2
3    sns.scatterplot(ax=eixos[0], x=dframe['xcoords'],
4                                 y=dframe['cosseno'])
5    eixos[0].set_title('Cosseno')
6
7    sns.lineplot(ax=eixos[1], x=dframe['xcoords'],
8                              y=dframe['seno'])
9    eixos[1].set_title('Seno')
```

A função `subplot()` pode criar vários eixos para desenhar gráficos independentes, como ilustra a Figura 2.12, criado pelo trecho de programa 2.5. Você pode copiar e executar esse trecho em um novo bloco do Jupyter Notebook. Observe que na linha 1 a função `subplot()` recebe 3 valores, o número de eixos horizontais, o número de eixos verticais e o tamanho da figura `figsize` (nesse caso, a unidade do tamanho é em polegadas). A função portanto cria dois lugares (eixos), lado a lado, para desenhar gráficos.

As linhas 3 e 7 desenham as funções cosseno e seno, respectivamente, usando formas diferentes de desenho (funções de plotagem). Os parâmetros `ax` de cada chamada de desenho indicam uma posição (eixo) criada pela `subplot()` na figura `fig2`.

O Seaborn (e o Matplotlib) permite a criação de gráficos de vários tipos distintos, que podem ser classificados em 3 categorias: gráficos relacionais (como o do seno e cosseno exibidos nessa seção), gráficos de distribuições (como histogramas) e gráficos de categorias (como o gráfico de barras usado para exibir médias). Uma forma até divertida para aprender a gerar esses gráficos é dando uma olhada na galeria de exemplos (https://seaborn.pydata.org/examples) do Seaborn para procurar algum gráfico que você acha apropriado para exibir os seus dados e então copiar e adaptar o trecho de código em um bloco do Jupyter Notebook.

2.4.5. Cálculo de ANOVA de 1 variável para um experimento intra-sujeitos

Agora podemos descrever como calcular uma ANOVA de 1 variável para o exemplo do estudo de teclados virtuais usando Python. Os passos para calcular o valor de F foram

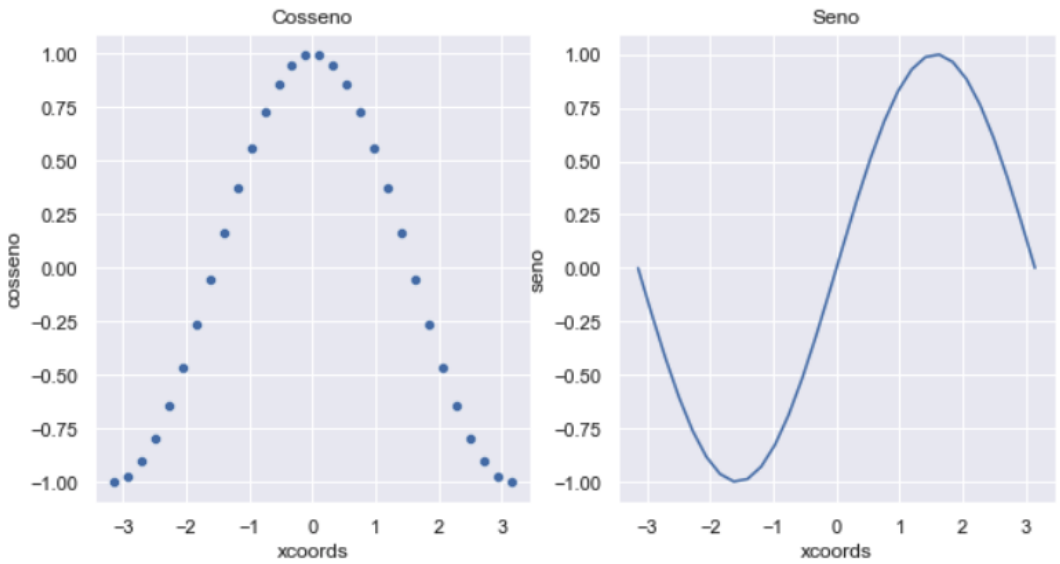

Figura 2.12. Exemplo de uso do `subplot()` para mostrar os gráficos isolados na mesma figura.

apresentados na Seção 2.3.1. Recorde que no experimento foram coletados dados de 10 participantes que usaram os dois teclados: sem aceleração e AugKey. Para cada participante foi calculada a média do desempenho com cada teclado e armazenados no arquivo "dados_teclados.csv". Você pode acessar esse arquivo junto ao material disponibilizado na página desse curso. O trecho de código Python abaixo cria as listas `A` e `B` com os mesmos valores dessa tabela, para que você possa copiar em seu programa. Fica como exercício transformar essas listas em um DataFrame. Esses são os mesmos valores usados na Seção 2.3.1 para o Exp1, ilustrados abaixo na forma de listas em Python.

```
A = [5.3, 3.5, 5.1, 3.6, 4.6, 4.1, 4.0, 4.8, 5.2, 5.1]
B = [5.7, 4.8, 5.1, 4.6, 6.1, 6.8, 6.0, 4.6, 5.5, 5.6]
```

Você pode ainda criar seu próprio arquivo .CSV com esses valores e, a seguir, abrir o arquivo e carregar os dados em um Pandas DataFrame, como mostrado no trecho de código 2.6.

Programa 2.6. Código para carregar os dados do experimento dos teclados em um DataFrame

```
1 df = pd.read_csv('dados_teclados.csv', sep=';', index_col=0)
2 print(df)
```

Para calcular a soma dos quadrados total, entre as condições e entre as pessoas participantes, precisamos calcular a média e a soma de todos os dados. Para isso usamos a função "flatten" que transforma os dados organizados em colunas em uma lista só, como mostrado no trecho de código 2.7.

Programa 2.7. Código para calcular um termo comum usado em todas as expressões de soma de quadrados

```
# Precisamos de media e a soma total de todos os dados
all_data_list = df.to_numpy().flatten()
mean_all = np.mean(all_data_list)
sum_all = np.sum(all_data_list)

# Este termo e subtraido de cada uma das somas de quadrados
common_term = sum_all**2 / (df.shape[0] * df.shape[1])
```

Agora calculamos a soma dos quadrados total, entre condições e entre participantes, como mostrado no trecho de código 2.8.

Programa 2.8. Código para calcular as somas dos quadrados usadas no cálculo de F

```
# Soma dos quadrados total
SST = sum(np.power(all_data_list, 2)) - common_term

# Soma dos quadrados entre as condicoes
SSB = sum(df.sum(axis=0).pow(2))/df.shape[0] - common_term

# Soma dos quadrados entre os participantes
SSS = sum(df.sum(axis=1).pow(2))/df.shape[1] - common_term

# Soma dos quadrados do erro
SSE = SST - SSS - SSB
```

Finalmente, calculamos os graus de liberdade e as médias da soma dos quadrados necessários para calcular o valor de F, como mostrado no trecho de código 2.9. No final da execução, o valor de F (arredondado) deve ser igual a 10.5.

Programa 2.9. Código para calcular aos graus de liberdade, a média dos quadrados e o valor de F

```
# Graus de liberadade de cada soma
SSS_df = df.shape[0]-1
SSB_df = df.shape[1]-1
SSE_df = SSS_df*SSB_df

# Calculando as medias da soma dos quadrados
MSb = SSB / SSB_df
MSs = SSS / SSS_df
MSe = SSE / SSE_df

# Finalmente, calculamos o valor de F
F = MSb / MSe
```

2.4.6. Análise estatística usando Scipy

Por fim vamos introduzir o Scipy, um módulo de código aberto em Python que possui muitas ferramentas úteis para o processamento de dados científicos e complementam os

outros módulos já citados como Numpy e Seaborn.

Vamos ver um exemplo de como calcular a regressão linear usando Scipy, que é um recurso que vamos utilizar na próxima seção. A regressão linear permite estimar a relação linear entre dois conjuntos de dados: um de entrada X e um de saída Y. Vamos mostrar um exemplo de como fazer uma regressão linear usando Scipy e mostrar os resultados com Seaborn. A Figura 2.13 mostra um trecho de código com as listas `X` e `Y` que serão utilizadas como entrada para regressão linear, além do gráfico com a distribuição desses dados.

```
X=[2.45, 5.96, 3.58, 3.78, 0.24, 2.38, 3.87, 6.80, 8.39, 7.6]
Y=[10.8, 21.5, 20.3, 20.4, 2.79, 13.4, 14.2, 27.76, 39.1, 29.6]

sns.scatterplot(x=X, y=Y)
```

```
<AxesSubplot:>
```

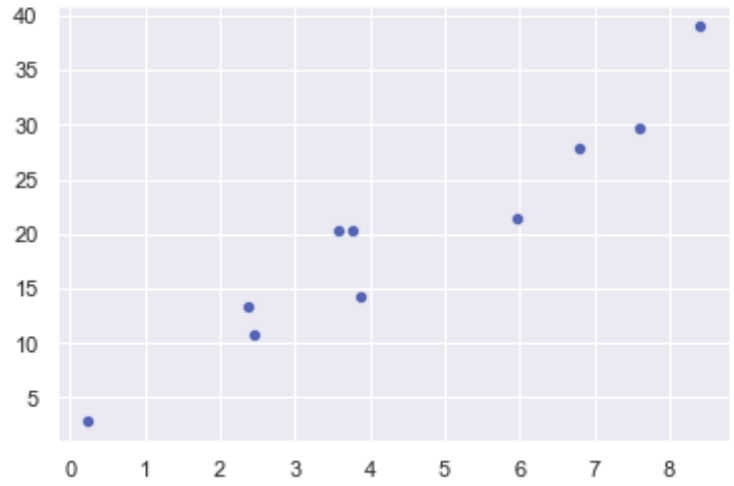

Figura 2.13. Distribuição dos dados de entrada nas listas X e Y usados para regressão linear.

A seguir vamos importar o módulo ”stats” do Scipy e rodar a regressão linear. O resultado da regressão retorna, entre outros, o valor da intersecção e a inclinação da reta obtida pela regressão linear, assim como o valor de R^2. O valor de R^2 varia entre 0 e 1 e representa quão bom a reta resultante da regressão se ajusta aos dados de entrada (quanto mais perto de 1 melhor o resultado).

Programa 2.10. Preparando dados para regressão linear

```
1     from scipy import stats
2
3     regressao = stats.linregress(X, Y)
4
5     slope = regressao.slope
6     inter = regressao.intercept
7     R2 = regressao.rvalue**2
8
9     print("Slope: {:0.2f}".format(slope))
10    print("Intersecao: {:0.2f}".format(inter))
11    print("Ajuste R2: {:0.2f}".format(R2))
```

O resultado do trecho de código 2.10 é o seguinte:

```
Slope: 3.84
Interseção: 2.69
Ajuste R2: 0.92
```

Agora podemos mostrar a reta resultante da regressão nos dados de entrada, para ver quão perto cada ponto ficou da reta. O trecho de código no topo da Figura 2.14 pode ser usado para exibir os resultados no mesmo gráfico. Para desenhar a reta usamos os valores mínimo e máximo de X e calculamos os pontos correspondentes de acordo com os parâmetros da reta que obtivemos na regressão linear. Na figura podemos observar também como a reta obtida passa próximo da maioria dos pontos usados na regressão linear.

2.5. Parte Experimental

O objetivo dessa seção é realizar um experimento prático, onde poderão ser aplicados os conhecimentos teóricos apresentados no curso e usar as ferramentas estudadas para análise de dados. Trata-se de um típico experimento para avaliar o desempenho humano em uma tarefa simples, para levantar uma curva de desempenho conhecida como lei de Fitts [Fitts 1954].

2.5.1. Motivação

Muitas tarefas realizadas no dia a dia exigem movimentos mecânicos das nossas mãos, braços, dedos, pés, etc. Por exemplo, quando digitamos em um teclado de computador, apontamos e clicamos objetos com o mouse, ou ainda quando utilizamos um telefone celular para enviar uma mensagem de texto, navegar pela internet ou brincar com algum jogo eletrônico.

Suponha que queremos desenvolver um novo apontador laser para substituir o apontamento pelo mouse convencional. Algumas perguntas que podemos estar interessados em responder são:

- Como saber se o novo dispositivo será mais rápido e eficiente para apontamento comparado com o o mouse?

```
X_reg = [min(X), max(X)]
Y_reg = [X_reg[0]*slope + inter, X_reg[1]*slope + inter]

sns.scatterplot(x=X, y=Y)
sns.lineplot(x=X_reg, y=Y_reg)
```

```
<AxesSubplot:>
```

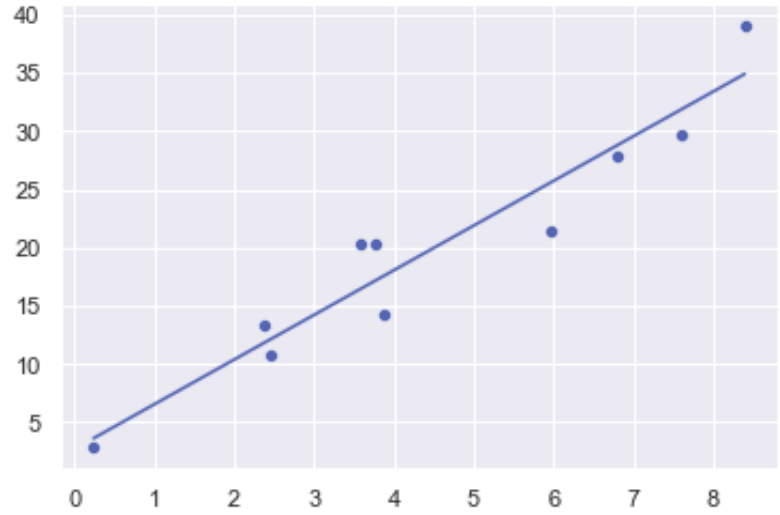

Figura 2.14. Trecho de código e correspondente resultado da regressão linear. Observe como a maioria dos pontos usados na regressão linear estão bem próximos da reta.

- O novo dispositivo é mais rápido para apontar objetos que estão mais próximos entre si ou mais afastados?
- Qual é a relação entre o tamanho dos alvos e o tempo de apontamento no novo dispositivo?

2.5.2. Experimento

Para responder cada uma dessas (e outras) perguntas, podemos fazer um experimento distinto. Por exemplo, podemos exibir alvos espalhados em um monitor, e pedir para uma pessoa voluntária que clique usando um dos dispositivos no alvo ”vermelho”, como mostrado na Figura 2.15. Após clicar em um alvo, um novo alvo se torna vermelho e deve ser clicado. O experimento pode consistir em clicar vários alvos e medir o tempo de seleção entre dois ”cliques”consecutivos.

Nesse caso, o tempo médio de seleção por participante poderia ser utilizado para comparar o desempenho dos dois dispositivos usados para apontamento. A análise dos resultados usaria ANOVA com um fator (mouse ou laser).

Você mesmo pode fazer um experimento assim, por exemplo, recortando alguns pedaços de papel e escrevendo números sobre eles. Ou ainda, simplificando a tarefa para dois alvos apenas, e medindo o tempo para tocar em cada um, alternadamente, por

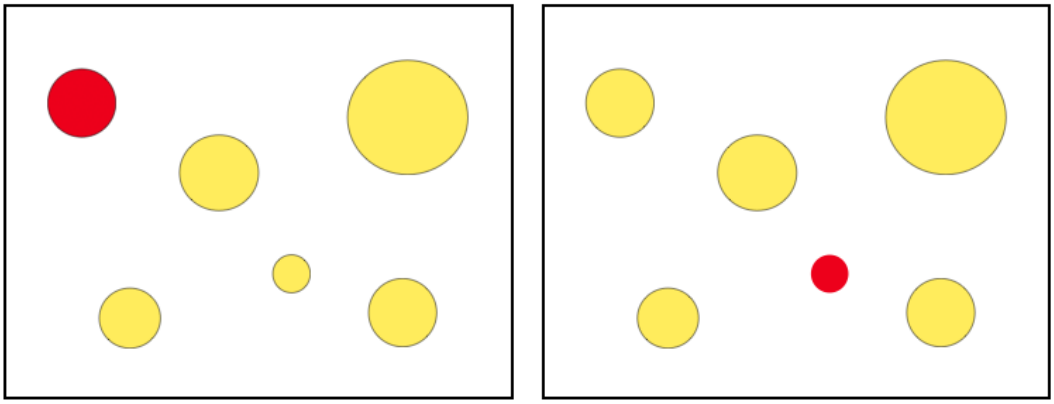

Figura 2.15. Tarefa típica de apontamento. Considere a tarefa de clicar no alvo vermelho na tela da esquerda. Após selecionar o alvo, um novo alvo vermelho é exibido e deve ser selecionado (clicado na sequência).

um certo número de vezes (como 10 ou 20 vezes). Peça para outras pessoas fazerem essa tarefa algumas vezes (3 ou 4 por exemplo) usando cada mão (esquerda e direita), enquanto você mede o tempo com um cronômetro e anota os resultados. Depois de coletar os dados, pode usar as ferramentas computacionais apresentadas na seção anterior para comparar o tempo de apontamento entre dois tipos distintos de apontamento (fator), que correspondem à mão direita e à mão esquerda. Talvez seja ainda melhor comparar o desempenho da mão dominante e não dominante, para considerar destros e canhotos.

A segunda e terceira perguntas indicam a possibilidade de haver outros fatores envolvidos e que podem afetar a velocidade de apontamento. Afinal, para apontar é necessário mover o cursor de uma certa distância (portanto quanto maior a distância menor deve ser a velocidade) e também parece ser mais difícil posicionar o cursor sobre um alvo pequeno. Parece intuitivo que, quanto menor o alvo, mais difícil é atingir ou tocar no alvo.

Novamente, podemos fazer mais experimentos com apenas 2 alvos, aumentando a distância e também variando o tamanho do papel usado como alvo. Por exemplo, poderíamos usar 3 tamanhos de alvo e 3 distâncias diferentes, resultando em 18 condições distintas: 2 métodos, 3 tamanhos e 3 distâncias. Repare como a complexidade do experimento aumenta com o número de fatores. Se cada participante do experimento precisar repetir uma condição 3 vezes, cada pessoa teria de fazer 54 tarefas de apontamento ao todo. A análise dos resultados também se torna mais complicada devido ao aumento do número de fatores e das possíveis interações entre eles.

Além de ajudar a testar hipóteses (responder as perguntas de pesquisa), outro objetivo de uma pesquisa pode ser comprovar uma teoria ou modelo. Note que esse é um objetivo bem mais ambicioso e que tem um maior potencial de contribuição científica também, como a Teoria da Relatividade de Einstein ou da Gravitação de Newton.

Mas não precisamos ser tão ambiciosos ou ir tão longe assim. Podemos tentar construir modelos (matemáticos) para prever o comportamento da resposta a certas

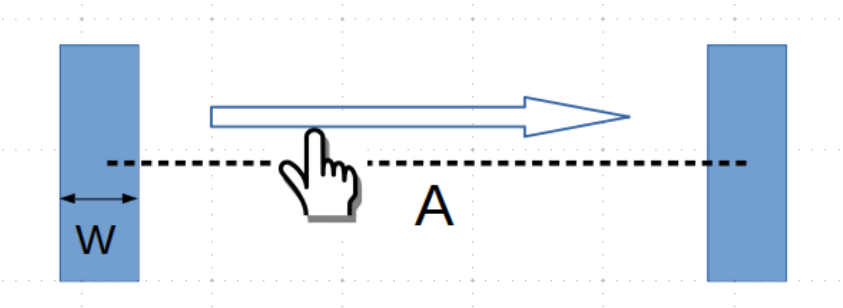

Figura 2.16. Exemplo onde o tempo de movimento do dedo indicador entre os alvos de largura W a uma distância A pode ser modelado pela lei de Fitts

variações dos fatores. Nesse sentido, a lei de Fitts [Fitts 1954] é um excelente exemplo muito utilizado em IHC que define um índice de dificuldade para uma tarefa de apontamento baseado na distância e no tamanho dos alvos, e pode ser usado para prever o desempenho humano sob certas condições.

2.5.3. Lei de Fitts

A lei de Fitts permite modelar o tempo de movimento MT entre alvos de largura W colocados a uma distância A (também chamada de amplitude do movimento), como é mostrado na Figura 2.16.

Para estimar o valor de MT, Fitts definiu antes o **índice de dificuldade** como sendo:

$$\mathrm{ID} = log_2(\frac{2A}{W}). \tag{1}$$

O tempo de movimento MT é definido com base no índice de dificuldade ID conforme a equação 2:

$$\mathrm{MT} = a + b * \mathrm{ID}. \tag{2}$$

Posteriormente, Mackenzie [MacKenzie 1992] sugeriu o uso de uma formulação semelhante à entropia de Shannon [Shannon and Weaver 1998] definida pela equação:

$$\mathrm{ID} = log_2(1 + \frac{A}{W}), \tag{3}$$

que oferece uma aderência melhor ao desempenho humano típico usando um mouse para apontamento, além de garantir que o índice de dificuldade é sempre um número não negativo.

Vamos ver agora como podemos interpretar, calcular os parâmetros e aplicar a Lei de Fitts para prever o desempenho humano em certas tarefas de apontamento. Para tal fim usaremos o índice de dificuldade como apresentado na equação 3.

Como podemos observar na equação 2, o tempo de movimento depende de dois valores a e b que são desconhecidos. O que eles representam e como podemos calculá-los? Se observarmos bem, o tempo de movimento é representado pela equação de uma

reta (equação linear ou de primeira ordem), onde o eixo X representa os valores do índice de dificuldade e o eixo Y representa o tempo de movimento.

Como já sabemos, em uma equação linear o valor de a, que é o termo independente, representa a intersecção da reta com o eixo Y. Quando o índice de dificuldade é zero, o valor de a é igual ao tempo de movimento (pois o termo $b * \mathbf{ID}$ é anulado). Como o valor de ID é sempre não negativo, podemos afirmar que o valor de a na equação 2 é o tempo mínimo possível de movimento. Mas porquê podemos afirmar que o ID é sempre não negativo? Analisando a equação 3 temos que o termo A/W é sempre não negativo, pois não faria muito sentido tocar ou clicar em um alvo de largura menor ou igual a zero (pois nesse caso o alvo seria invisível), assim como também não faria sentido pensar em distância entre alvos como sendo menor ou igual a zero, pois os alvos estariam sobrepostos. Logo temos que $log_2(1 + \frac{A}{W})$ será sempre não negativo.

O valor de b representa a inclinação da reta, ou seja, como varia o tempo de movimento em função do índice de dificuldade **ID**. Um valor positivo de b significa que a cada 1 unidade de incremento do índice de dificuldade, o tempo de movimento é acrescentado de b unidades. Já um valor negativo de b significa que um incremento no índice de dificuldade produz uma redução no tempo de movimento. Quanto maior for o valor absoluto de b, maior a sensibilidade do tempo de movimento em função do índice de dificuldade.

Agora que já sabemos interpretar os valores de a e b, como eles podem ser calculados? Uma alternativa é estimar esses valores de forma empírica, ou seja, medindo o comportamento real de várias pessoas, podemos estimar com uma boa precisão os valores de a e b para uma dada população.

Para isto precisamos calcular experimentalmente o tempo de movimento para vários valores de ID, afim de obter vários pontos para estimar os parâmetros da reta. Ou seja, precisamos avaliar o desempenho usando vários valores de ID, medindo o tempo MT para cada ID e depois usar uma técnica conhecida como regressão linear para estimar os valores de a e b.

A Figura 2.17 mostra os dados de tempo de movimento para vários valores de ID coletados de um participante. O eixo X representa os valores de ID e o eixo Y os valores de tempo de movimento medidos em milissegundos. Estes dados foram coletados durante um aula experimental sobre a lei de Fitts e, portanto, representam dados reais de um dos participantes do experimento.

Uma questão que surge é como definir os valores de ID. Como vimos na equação 3 o índice de dificuldade depende da largura dos alvos W e da distância entre eles A, que são as variáveis independentes do estudo. Então podemos escolher vários valores para W e A que façam sentido na vida real, que reflitam cenários onde os dispositivos de apontamento seriam usados. Como no exemplo que estamos estudando o apontamento foi realizado com o dedo sobre uma mesa, os valores de largura dos alvos foram 2, 5 e 10 cm e as distâncias entre os alvos foram 5, 10, 20 e 40 cm.

Uma vez definidos os valores de W e A, calculamos todas as combinações dessas variáveis independentes, que caracteriza um experimento com **desenho fatorial**, onde para cada nível de uma variável incluímos todos os níveis da outra. Para gerar todas as combinações de W e A escolhemos, para valor possível de W, todos os valores possíveis

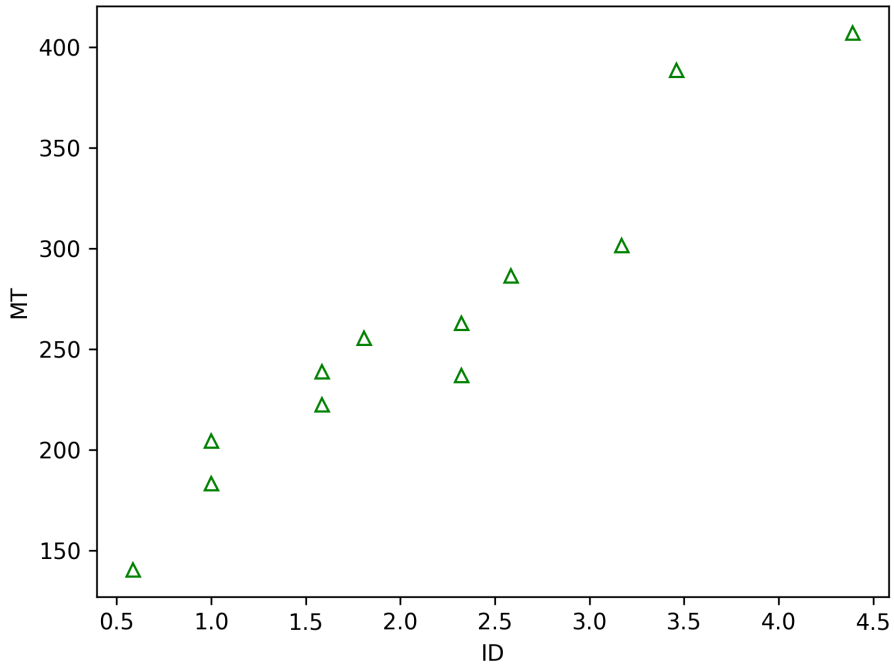

Figura 2.17. Exemplo de coleta do tempo de movimento para vários valores de índice de dificuldade para um participante.

de A. Isto daria no total $3 \times 4 = 12$ configurações possíveis. A Tabela 2.4 mostra todas as 12 configurações possíveis do estudo e a média do tempo de movimento para cada uma. Tal média foi calculada a partir de várias repetições para cada configuração experimental.

Como podemos notar na Tabela 2.4 o mesmo ID pode ser obtido a partir de mais de um par de valores de W e A. Esta situação não invalida o estudo realizado, pois existe um motivo por trás de tal repetição, que é a escolha de valores de W e A que fazem sentido na vida real. A escolha de outros valores de W e A de forma que o valor de ID não apareça repetido também seria um desenho válido.

O método de regressão linear fornecido pela pacote **Stats** da biblioteca **Scipy**, que faz parte do **Numpy**, recebe dois parâmetros de entrada: um vetor n-dimensional X com os valores das variáveis independentes (cada coluna corresponde a uma variável) e um vetor unidimensional Y com os valores da variável dependente.

No nosso exemplo temos apenas uma variável independente: o índice de dificuldade. Já a variável dependente é o tempo de movimento. O resultado da regressão linear com os dados de exemplo é mostrado na Figura 2.18. O valor de R^2 obtido foi de 0.93, lembrando que quanto mais perto de 1, melhor a reta resultante da regressão se ajusta aos dados de entrada.

Os coeficientes a e b obtidos da regressão linear são 117.9 e 66.4 respectivamente. Assim, a equação da lei de Fitts para os dados do nosso exemplo resultante é:

$$\mathbf{MT} = 117.9 + 66.4 \times \mathbf{ID} \tag{4}$$

Tabela 2.4. Configurações experimentais de largura dos alvos, amplitude do movimento, índice de dificuldade e tempo de movimento coletado experimentalmente.

Largura W	Amplitude A	Índice de dificuldade ID	Média do tempo de movimento MT
2	5	2.32	225.5
	10	3.32	286.5
	20	4.32	388.5
	40	5.32	407
5	5	1	204.5
	10	2	239
	20	3	263
	40	4	301.5
10	5	0	140.4
	10	1	183.5
	20	2	222.5
	40	3	407.5

O valor de a=117.9 é o tempo mínimo de movimento possível. Podemos entender esse valor como sendo o limite fisiológico, considerando que o nosso corpo precisa de um tempo para perceber o estímulo, transmitir a informação até o cérebro e acionar os músculos, que trabalham coordenadamente para completar a tarefa. Já o valor de b=66.4 significa que a cada incremento de 1 unidade no índice de dificuldade, o tempo de movimento aumenta em 66.4 ms.

O que acontece se queremos estimar o tempo de movimento para outros valores de ID não considerados inicialmente no experimento?

Vamos supor que queremos saber como seria o tempo de movimento para alvos de largura 5 cm colocados a uma distância de 30 cm. Primeiro devemos calcular o ID conforme a equação 3:

$$\mathbf{ID}_{teste} = log_2(1 + \frac{30}{5}) = 2.8. \tag{5}$$

Agora calculamos o tempo de movimento:

$$\mathbf{MT}_{teste} = 117.9 + 66.4 \times 2.8 = 304.3ms. \tag{6}$$

A Figura 2.19 mostra o novo ponto estimado usando a expressão da lei de Fitts.

2.6. Onde chegamos e para onde você pode ir

Nesse curso apresentamos brevemente vários tópicos relacionados ao método científico, análise estatística, e ferramentas computacionais para a exploração de dados. Nosso propósito foi acompanhar você numa rápida visita a alguns desses tópicos para, quem sabe, atrair você para fazer algumas visitas futuras, mais prolongadas, que conduzam você a realizar estudos com usuários que venham a enriquecer ainda mais a área de IHC e contribuir em alguma inovação tecnológica. Para se aprofundar nesses tópicos recomendamos a leitura do livro do Prof. Scott Mackenzie [Mackenzie 2012].

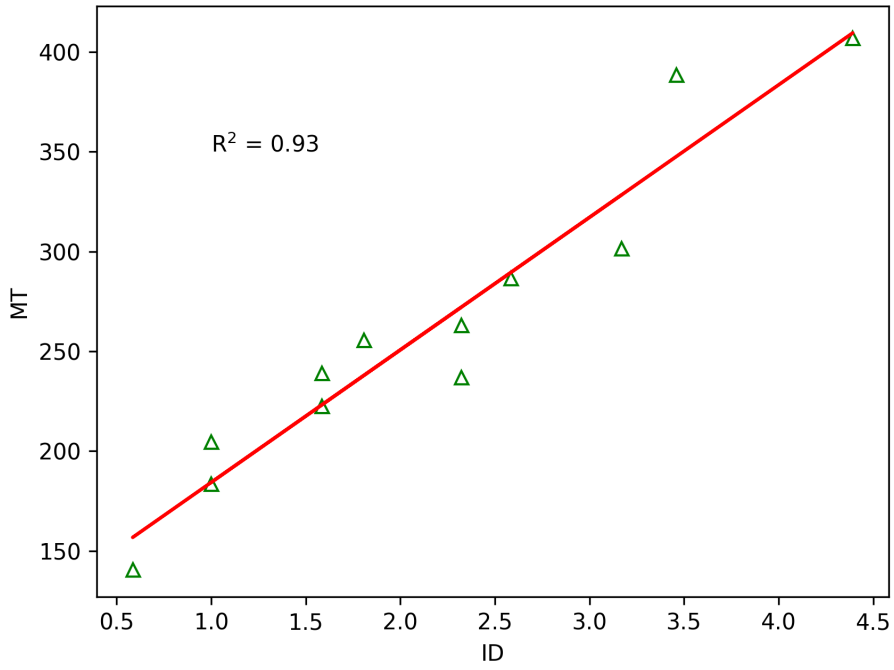

Figura 2.18. Resultado da regressão linear mostrando a reta $MT = 117.9 + 66.4 \times ID$ para os dados de exemplo.

Nesse curso descrevemos como aplicar o método científico no projeto de experimentos com usuários de algum sistema computacional, além de algumas ferramentas computacionais existentes para a análise e exploração de dados na linguagem Python. Vimos que o projeto de qualquer experimento envolvendo humanos deve considerar valores éticos desde o início, devendo inclusive ser aprovado por um comitê de ética independente para que o experimento possa ser realizado. No Brasil, a avaliação e acompanhamento dos experimentos é realizado pelo Ministério da Saúde. Na área de IHC, vimos que uma grande maioria de experimentos busca utilizar dados quantitativos devido a sofisticação do conhecimento que resultados quantitativos podem apresentar em relação a dados qualitativos. Isso não diminui, no entanto, a importância de dados qualitativos nas pesquisas.

Seja quantitativo ou qualitativo, há vários testes estatísticos que podem ser aplicados para comprovar ou não a hipótese de pesquisa sendo investigada. Nos utilizamos o AugKey [Diaz-Tula and Morimoto 2016], um teclado de realidade aumentada desenvolvido pelos autores para auxiliar pessoas com deficiência motora na tarefa de entrada de textos com o olhar, como exemplo para introduzir e discutir vários conceitos e problemas relacionados ao desenho experimental. Para conhecer outros exemplos de aplicação do método científico em IHC, recomendamos a leitura de outros trabalhos publicados no CHI (*Conference on Human Factors in Computing Systems*) organizado pela ACM (*Association for Computing Machinery*), uma das principais conferências internacionais na área de IHC e com grande foco em estudos com usuários. O IHC (Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais) organizado pela SCB (Sociedade Brasileira de Computação) também é uma excelente referência para trabalhos nacionais

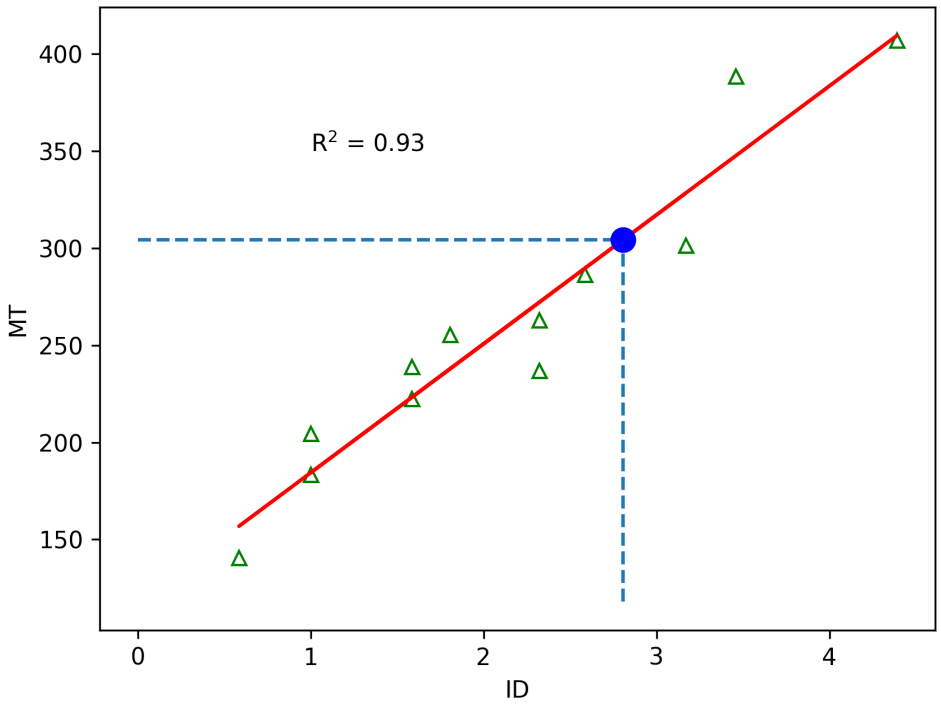

Figura 2.19. Estimando o tempo de movimento para um ponto não incluído no experimento, com largura 5 cm e distância de movimento 30 cm. O índice de dificuldade é de 2.8 e o tempo de movimento de 304.3 ms.

e, alguns deles, em português.

Um outro passo importante para a condução de experimentos é um bom conhecimento sobre as técnicas de análise paramétricas e não paramétricas. Dada a importância das técnicas paramétricas para analisar dados quantitativos e, em particular, da técnica de Análise de Variâncias (ANOVA) para a IHC, discutimos ANOVA com um pouco mais de cuidado para que você possa entender melhor os resultados apresentados em vários trabalhos na área.

Por fim, as ferramentas computacionais evoluem muito rapidamente e, apesar de existirem ferramentas mais específicas para análise estatística, achamos que apresentar um conjunto de ferramentas para análise e exploração de dados científicos baseadas na linguagem Python e no ambiente Jupyter Lab facilite o entendimento dos exemplos e a rápida adequação das ferramentas às suas necessidades. Todas as ferramentas apresentadas, como Numpy, Seaborn, Pandas e Scipy, são muito poderosas e merecem um visita futura mais cuidadosa, caso você tenha se interessado por alguma. Todas as ferramentas utilizada também são de domínio público (*open source*) e podem ser baixadas e utilizadas gratuitamente. Todos os exemplos apresentados nesse curso estão disponíveis no endereço https://www.ime.usp.br/~hitoshi/jai2021.

Referências

[Buxton et al. 1985] Buxton, W., Hill, R., and Rowley, P. (1985). Issues and techniques in touch-sensitive tablet input. *SIGGRAPH Comput. Graph.*, 19(3):215–224.

[Diaz-Tula and Morimoto 2016] Diaz-Tula, A. and Morimoto, C. H. (2016). Augkey: Increasing foveal throughput in eye typing with augmented keys. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 3533–3544, New York, NY, USA. ACM.

[Fitts 1954] Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental PSychology*, 74:381–391.

[Kaptein et al. 2010] Kaptein, M., Nass, C., and Markopoulos, P. (2010). Powerful and consistent analysis of likert-type rating scales. In *Proceedings of the ACM SIGCHI Conference on Human-Factors in Computing Systems—CHI 2010*, page 2391–2394, New York, NY, USA. Association for Computing Machinery.

[Lazar et al. 2017] Lazar, J., Feng, J. H., and Hocheiser, H. (2017). *Research Methods in Human-Computer Interaction*. John Wiley & Sons.

[Mackenzie 2012] Mackenzie, I. (2012). *Human-Computer Interaction: An Empirical Research Perspective*. Morgan Kauffmann Publishers, New York.

[MacKenzie 1992] MacKenzie, I. S. (1992). Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction*, 7(1):91–139.

[Mackenzie et al. 1999] Mackenzie, I. S., Zhang, S. X., and Soukoreff, R. W. (1999). Text entry using soft keyboards. *Behaviour & Information Technology*, 18(4):235–244.

[Majaranta 2009] Majaranta, P. (2009). *Text Entry by Eye Gaze*. PhD thesis, University of Tampere, Department of Computer Science, Tampere, Finland.

[Morimoto and Mimica 2005] Morimoto, C. H. and Mimica, M. (2005). Eye gaze tracking techniques for interactive applications. *Computer Vision and Image Understanding*, 98:4–24.

[Nielsen 1994] Nielsen, J. (1994). Enhancing the explanatory power of usability heuristics. In *Conference Companion on Human Factors in Computing Systems*, CHI '94, page 210, New York, NY, USA. Association for Computing Machinery.

[Robertson and Kaptein 2018] Robertson, J. and Kaptein, M. E. (2018). *Modern Statistical Methods for HCI*. Springer.

[Selker 2008] Selker, T. (2008). Touching the future. *Commun. ACM*, 51(12):14–16.

[Shannon and Weaver 1998] Shannon, C. and Weaver, W. (1998). *The Mathematical Theory of Communication*. University of Illinois Press.

[Sharp et al. 2019] Sharp, H., Preece, J., and Rogers, Y. (2019). *Interaction Design: Beyond Human-Computer Interaction*. Wiley, 5th edition.

[Urbina and Huckauf 2010] Urbina, M. H. and Huckauf, A. (2010). Alternatives to single character entry and dwell time selection on eye typing. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, ETRA '10, pages 315–322, New York, NY, USA. ACM.

[Wobbrock et al. 2008] Wobbrock, J. O., Rubinstein, J., Sawyer, M. W., and Duchowski, A. T. (2008). Longitudinal evaluation of discrete consecutive gaze gestures for text entry. In *Proceedings of the 2008 symposium on Eye Tracking Research & Applications*, ETRA '08, pages 11–18, New York, NY, USA. ACM.