A Pragmatic Introduction to Some Common Analyses

CSE 510 – Advanced Topics in Human-Computer Interaction

DUE: Sunday, May 14

Description

You will gain basic familiarity with analyzing experiments using mixed-model analyses of variance in either the JMP or R statistical packages. Consistent with lecture, this assignment is not intended to provide complete knowledge of how to design or analyze experiments (a topic far beyond the scope of one lecture or assignment). This assignment is instead focused on a pragmatic introduction to analyzing experiments based in designs you might later find useful. Please consider this assignment in the context of the material covered in lecture, as not all of it is repeated here.

In addition to my lecture material and the contents of this assignment, you might benefit from working through the first four sections of *Practical Statistics for Human-Computer Interaction*, an independent study created by Jacob Wobbrock and linked from the course webpage:

https://depts.washington.edu/aimgroup/proj/ps4hci/ps4hci.zip

The first three sections provide an introduction to basic statistical concepts, how to interpret data, and analyses of variance. These sections require you to do some independent research in order to complete the questions (e.g., using a statistics textbook). The fourth section is structured more as a tutorial and gives a solid introduction to several types of analyses, including mixed-model analyses (used in this assignment). Depending on your existing knowledge, you may be able to skip some or all of Sections 1 to 3 to focus only on the mixed-models portions of Section 4. An answer key is provided, and you are not required to hand in any work from this guide.

This assignment was originally developed using the JMP statistical package. R has since also become popular and mature. We highly recommend that you use these with an IDE for analysis. Images in the JMP version of this assignment are from Version 7, but the functionality is the same. You are also free to use any other package, but you will likely find this assignment much more difficult to complete and we will not be able to provide any assistance in completing it correctly.

JMP includes an IDE, and is available as a free trial, or through a UW CSE discount:

http://www.jmp.com/en_us/software/jmp.html

https://e5.onthehub.com/WebStore/AdTargetOfferingList.aspx?wsmv=7ed98060-98ea-e411-940b-b8ca3a5db7a1&ws=a4fce2bc-ac2d-de11-a497-0030485a8df0&vsro=8&utm_source=jmp-version12-rs&utm_medium=WebStoreAds&utm_campaign=JMP

RStudio is a free IDE available at:

https://www.rstudio.com/

Working it Through with a Partner

When analyzing data (e.g., to write a paper), it is often valuable to talk through your analyses with another person. This is useful for checking that what you did sounds correct and for thinking about how to proceed if stuck. You are welcome to work with a partner throughout this assignment.

Only one person in the partnership needs to submit an assignment. If you work with a partner, please include their name near the top of your report. Please indicate in your submission if you talked through the analysis with others who were not your partner.

Data Files and Formatting

You will work with three datasets: one artificial and two from actual published studies. The data is appropriate for these analyses, but explicitly not cleaned up for the sake of this assignment. The primary implication is that you need to be mindful of the types which your package assigns to columns when you load data files. It will be your responsibility to decide whether each field should be *continuous* or *nominal* (you can safely ignore the *ordinal* type for this assignment).

If using JMP, note you should likely save the provided CSV files into the JMP format, or your type information will be lost when you later come back to the file.

If using R, note R is an interpreted language. As a result, people tend to interleave scripting in a file and running commands on the command line. We have provided a sample script file to demonstrate the analyses in the artificially-generated dataset.

If using R, the easiest formatting solution is to use R Markdown or compile your R notebooks. This will create a PDF that contains your comments, scripting commands, and output rather than trying to copy and paste graphs, commands, etc. into a different document. R markup language will allow for the cleanest interleaving of the written components as well.

To compile your notebook in R Studio: File→Compile Notebook

To get started with R Markdown in R Studio: File→New File→R Markdown… will create a template with instructions to get you started.

Coordination and Submission Procedure

You should submit a report in PDF or HTML addressing the bulleted questions posed in the body of the assignment (items 1.1 through 1.6, 2.1 through 2.13, 3.1 to 3.2). Duplicate the bulleted question in your report, but not the explanatory text between questions. The assignment is available in both Word and PDF formats so that it is easier for you to duplicate the questions.

Be aware of the need to preserve high-resolution images in your electronic submission. If you submit a PDF including screenshots, ensure resolution is preserved so that we can zoom in to the point of being able to read any details. This is most likely to be a concern if you are outputting to PDF with settings that compress images.

Submit a PDF or a ZIP of your work via Canvas.

Grading

You will be graded on the *correctness* and the *appropriateness* of your responses to questions posed by the assignment. The notion of *correctness* is hopefully self-evident. Regarding *appropriateness*, grading will be based in striking an appropriate balance between reporting *sufficient* detail and reporting *excessive* detail. The goal is to gain experience reporting your results in approximately the same level of detail that should be included in reporting a research result.

Study 1: Text Input Method Words Per Minute

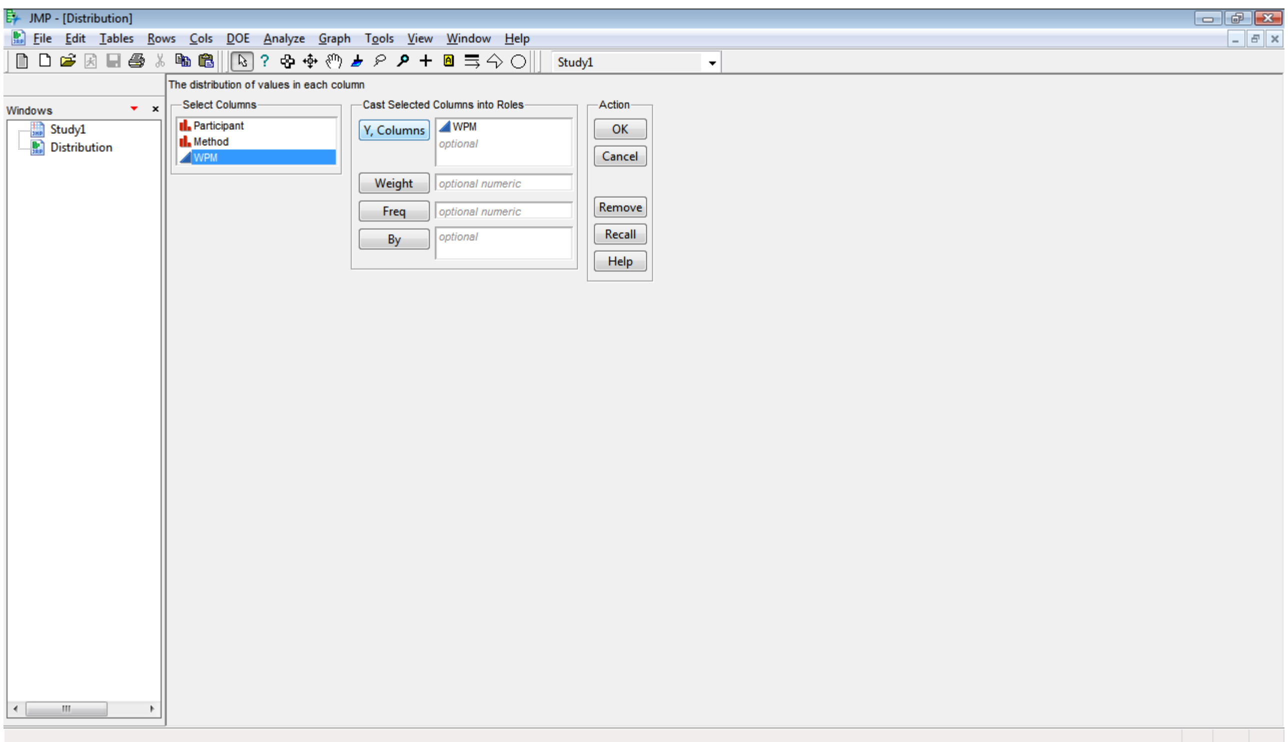
This is a small fictitious dataset from a previous version of Wobbrock's guide. It represents a within-subjects experiment measuring text input speed of ten participants using three different interfaces (speech, keyboard, and Graffiti). Wobbrock gives the following backstory for understanding this fictitious data:

> *The study compared three text input methods, Graffiti, keyboard typing, and speech recognition. After fifteen minutes of practice with each, subjects entered 20 phrases with each method. For each subject, trials whose WPM were less or more than 2 standard deviations from the subject's mean of trials were removed as outliers. The measure for each technique was the average of the non-outlier phrases for each method for each participant.*

Open the dataset in JMP. Assign appropriate variable types by clicking on the icons next to each variable name to the left of the matrix display.

1.1 What type did you assign to each of the variables. Briefly, why?

Now we will plot *WPM*. We do this both to get familiar with the dataset and to get familiar with capturing screenshots of our exploration and analysis of datasets in JMP. Access the Analyze menu, then choose Distribution. Plot the *WPM* variable by configuring the dialog like this:



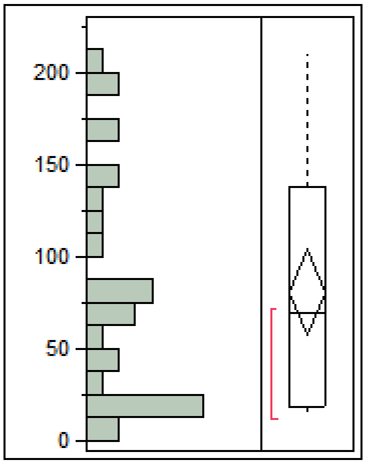
This screenshot of the full application was captured by pressing ALT-PRINTSCREEN on PC (CMD-SHIFT-4, then Space on Mac to select the window), which inserts the screenshot into the clipboard, then pasting the screenshot into this Word document. This will allow me (or in this case you) to later see exactly how the analysis was configured. You will need to do this as several points throughout this assignment, so go ahead and figure it out now.

1.2 [JMP] Insert a screenshot of your configuration of the dialog to plot the values of WPM.

You should get a result that looks like this. JMP has plotted the frequency of occurrence of different values of *WPM* and has computed some summary statistics, such as the mean *WPM*.

Distributions

WPM



Quantiles

100.0%	maximum	211.05
99.5%		211.05
97.5%		211.05
90.0%		186.47
75.0%	quartile	137.62
50.0%	median	69.78
25.0%	quartile	18.74
10.0%		13.31
2.5%		11.97
0.5%		11.97
0.0%	minimum	11.97

Moments

Mean	80.970667
Std Dev	63.199603
Std Err Mean	11.538616
upper 95% Mean	104.56979
lower 95% Mean	57.371547
N	30

Note how this capture is just the output of the analysis (not the surrounding window). It was captured through JMP's File → Save As functionality (saving as a PNG, then inserting that PNG into Word). This can be important in two ways. In this case, it means that the irrelevant aspects of the window have been automatically removed. In other cases, it can be helpful to capture the full output of an analysis even when that output does not fit on the screen (the screenshot functionality would capture only what is visible, but this approach captures everything accessible via scrolling). Be careful, because JMP will happily overwrite a previous file each time you do this. So insert the images into your Word file as you go, rather than trying to save them up (this is also obviously better for making sure that you do not make mistakes in documenting your process). Again, you will need to do this at several points throughout this assignment, so go ahead and figure it out now.

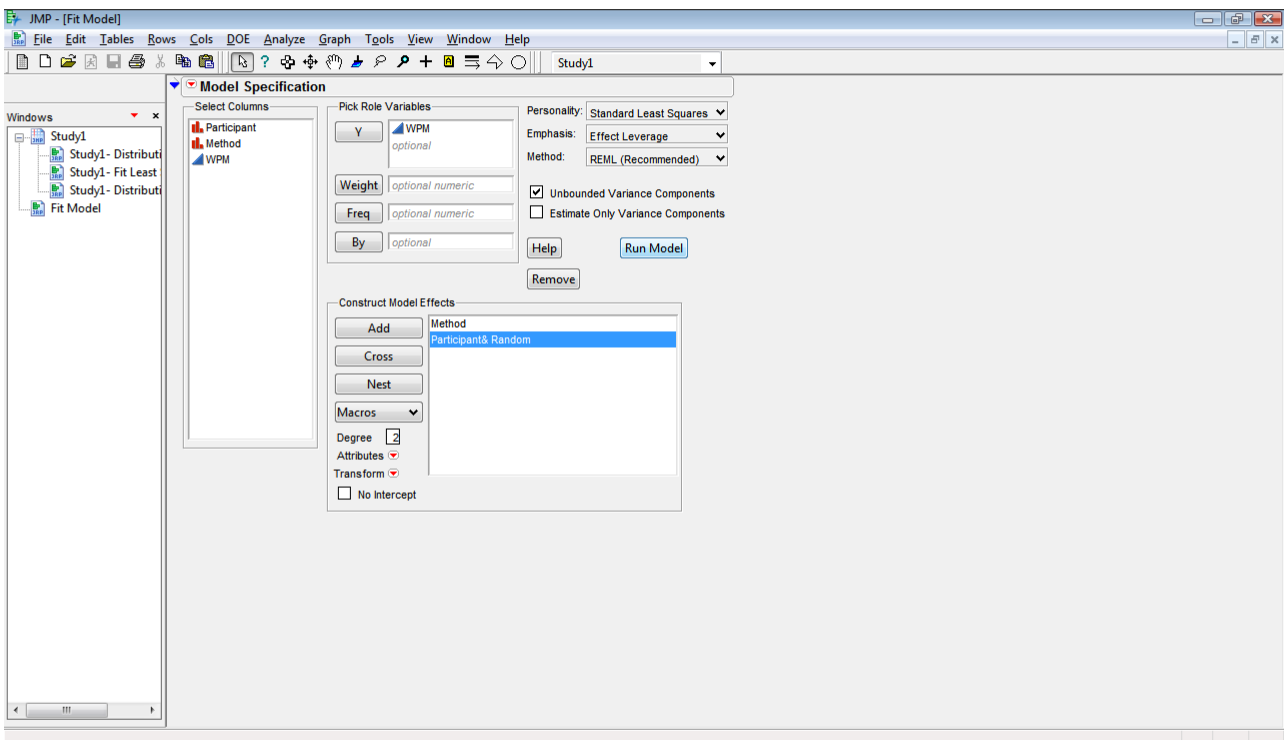
1.3 Insert a screenshot of plot of the values of WPM.

That plot does not seem to tell us much about the relationship of *WPM* to the different interfaces. So now we will split out *WPM* by interface. Close the existing plot and create a new one, this time adding *method* to the By field. JMP will split up the rows based on their value for *method*, showing three different plots. You should be able to look over this and see that different values of *method* seem to have different mean *WPM* values (at roughly 16, 68, and 158).

If *method* were the only thing that had impacted *WPM*, we might base our analysis on such a division of the data. But there is also a potential for *participant* to have impacted *WPM*, as some people might be slower or faster (regardless of which interface they are using). Plot the values of *WPM* by *participant*, and you can see that this does seem to be true (with participants having means ranging from as low as 65 to as high as 102). So our challenge is to analyze this data in a way that allows us to determine whether different *methods* actually enable faster input (or if the difference is instead simply due to individual variations).

In the language of mixed-model analyses, *method* is a fixed effect. It has three levels that were selected because they are of interest (we are interested in the effect of speech, keyboard, and Graffiti). In contrast, *participant* is a random effect. Participants were randomly sampled from a larger population over which we wish to generalize (we want to know whether *methods* are faster for the larger population) but we are not interested in whether P1 or P2 was faster.

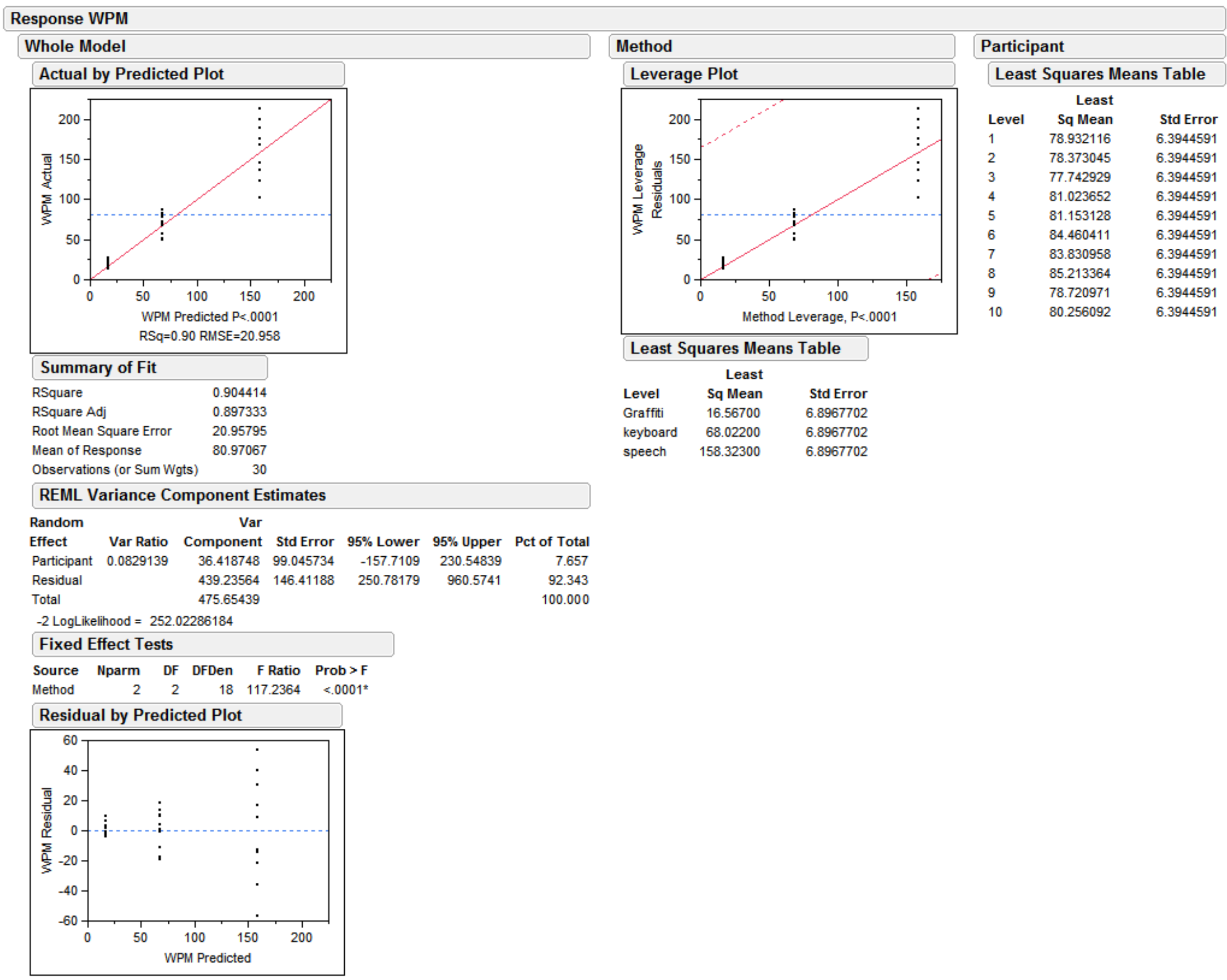
We therefore access the Analyze → Fit Model dialog and populate it like this:



By default, model effects are fixed. Note the “Participant& Random” indication that *participant* will be treated as a random effect. This was configured by adding *participant*, selecting it, and then selecting Random Effect from Attributes (click the little red and white thing).

1.4 Insert a screenshot of your configuration of the Run Model dialog.

Clicking Run Model will yield output like this:



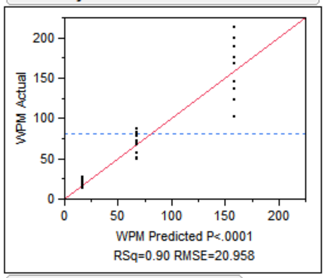
The first thing we will look at here is the section labeled “Fixed Effect Tests”. Here we can see there is just one variable listed. If we had accidentally forgotten to flag *participant* as a random effect, it would also appear here. This test shows us that *method* has a statistically significant impact on *WPM*. We can also see the least squares mean values for the different levels of both *method* and *participant*, but there has not yet been any test of whether these differences are significant. We will treat this as an unplanned comparison. We can access an appropriate test through the menu for the *method* variable, choosing the LSMeans Tukey HSD.

Adding the Tukey HSD gives us this:

Response WPM

Whole Model

Actual by Predicted Plot



Summary of Fit

RSquare	0.904414
RSquare Adj	0.897333
Root Mean Square Error	20.95795
Mean of Response	80.97067
Observations (or Sum Wgts)	30

REML Variance Component Estimates

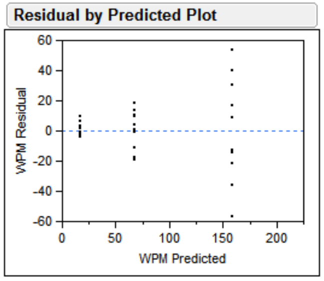
Random Effect	Var Ratio	Var Component	Std Error	95% Lower	95% Upper	Pct of Total
Participant	0.0829139	36.418748	99.045734	-157.7109	230.54839	7.657
Residual		439.23564	146.41188	250.78179	960.5741	92.343
Total		475.65439				100.000

-2 LogLikelihood = 252.02286184

Fixed Effect Tests

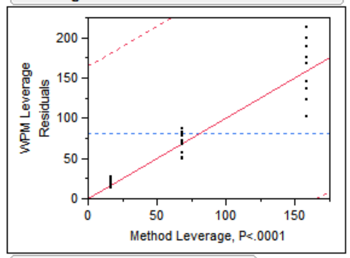
Source	Nparm	DF	DFDen	F Ratio	Prob > F
Method	2	2	18	117.2364	<.0001*

Residual by Predicted Plot



Method

Leverage Plot



Least Squares Means Table

Level	Least Sq Mean	Std Error
Graffiti	16.56700	6.8967702
keyboard	68.02200	6.8967702
speech	158.32300	6.8967702

LSMeans Differences Tukey HSD

a= 0.050

Mean[i]-Mean[j] / Std Err Dif / Lower CL Dif / Upper CL Dif (LSMean[i] rows, LSMean[j] columns)	Graffiti	keyboard	speech
Graffiti	0	-51.455	-141.76
	0	9.37268	9.37268
	0	-75.376	-165.68
	0	-27.534	-117.84
keyboard	51.455	0	-90.301
	9.37268	0	9.37268
	27.5344	0	-114.22
	75.3756	0	-66.38
speech	141.756	90.301	0
	9.37268	9.37268	0
	117.835	66.3804	0
	165.677	114.222	0

Level				Least Sq Mean
speech	A			158.32300
keyboard		B		68.02200
Graffiti			C	16.56700

Levels not connected by same letter are significantly different.

Participant

Least Squares Means Table

Level	Least Sq Mean	Std Error
1	78.932116	6.3944591
2	78.373045	6.3944591
3	77.742929	6.3944591
4	81.023652	6.3944591
5	81.153128	6.3944591
6	84.460411	6.3944591
7	83.830958	6.3944591
8	85.213364	6.3944591
9	78.720971	6.3944591
10	80.256092	6.3944591

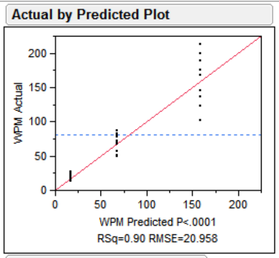
We can see by the fact that the levels are not connected that speech, keyboard, and Graffiti are all significantly different from each other (all three pairs of tests are significant). If we want to be able to fully report these differences, we need to add contrasts that will give us the full F-test information. Add these through the variable menu, giving levels opposing values that sum to 0.

Adding all three contrasts and expanding Test Detail to show the contrasts give us this:

Response WPM

Whole Model

Actual by Predicted Plot



Summary of Fit

RSquare	0.904414
RSquare Adj	0.897333
Root Mean Square Error	20.95795
Mean of Response	80.97067
Observations (or Sum Wgts)	30

REML Variance Component Estimates

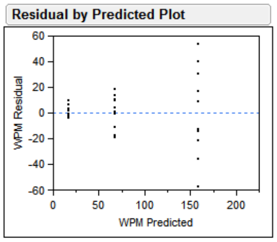
Random Effect	Var Ratio	Var Component	Std Error	95% Lower	95% Upper	Pct of Total
Participant	0.0829139	36.418748	99.045734	-157.7109	230.54839	7.657
Residual		439.23564	146.41188	250.78179	960.5741	92.343
Total		475.65439				100.000

-2 LogLikelihood = 252.02286184

Fixed Effect Tests

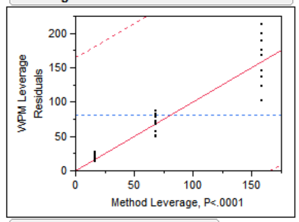
Source	Nparm	DF	DFDen	F Ratio	Prob > F
Method	2	2	18	117.2364	<.0001*

Residual by Predicted Plot



Method

Leverage Plot



Least Squares Means Table

Level	Least Sq Mean	Std Error
Graffiti	16.56700	6.8967702
keyboard	68.02200	6.8967702
speech	158.32300	6.8967702

LSMeans Differences Tukey HSD

a= 0.050

LSMean[i]	Mean[i]-Mean[j] / Std Err Dif / Lower CL Dif / Upper CL Dif	Graffiti	keyboard	speech
Graffiti	Mean[i]-Mean[j]	0	-51.455	-141.76
	Std Err Dif	0	9.37268	9.37268
	Lower CL Dif	0	-75.376	-165.68
	Upper CL Dif	0	-27.534	-117.84
keyboard	Mean[i]-Mean[j]	51.455	0	-90.301
	Std Err Dif	9.37268	0	9.37268
	Lower CL Dif	27.5344	0	-114.22
	Upper CL Dif	75.3756	0	-66.38
speech	Mean[i]-Mean[j]	141.756	90.301	0
	Std Err Dif	9.37268	9.37268	0
	Lower CL Dif	117.835	66.3804	0
	Upper CL Dif	165.677	114.222	0

Level		Least Sq Mean
speech	A	158.32300
keyboard	B	68.02200
Graffiti	C	16.56700

Levels not connected by same letter are significantly different.

Contrast

Test Detail

Graffiti	1
keyboard	-1
speech	0
Estimate	-51.46
Std Error	9.3727
t Ratio	-5.49
Prob>|t|	3.3e-5

Numerator DF	1
Denominator DF	18
F Ratio	30.138913596
Prob > F	0.0000325649

Contrast

Test Detail

Graffiti	1
keyboard	0
speech	-1
Estimate	-141.8
Std Error	9.3727
t Ratio	-15.12
Prob>|t|	1e-11

Numerator DF	1
Denominator DF	18
F Ratio	228.74695858
Prob > F	1.122576e-11

Contrast

Test Detail

Graffiti	0
keyboard	1
speech	-1
Estimate	-90.3
Std Error	9.3727
t Ratio	-9.634
Prob>|t|	1.6e-8

Numerator DF	1
Denominator DF	18
F Ratio	92.823416212
Prob > F	1.5790922e-8

Participant

Least Squares Means Table

Level	Least Sq Mean	Std Error
1	78.932116	6.3944591
2	78.373045	6.3944591
3	77.742929	6.3944591
4	81.023652	6.3944591
5	81.153128	6.3944591
6	84.460411	6.3944591
7	83.830958	6.3944591
8	85.213364	6.3944591
9	78.720971	6.3944591
10	80.256092	6.3944591

1.5 Insert a screenshot of your final dialog analyzing method and WPM.

This analysis was very straightforward, and all of the information you need to report it is included in your screenshots 1.4 and 1.5.

1.6 Close JMP and use your 1.4 and 1.5 screenshots to complete this paragraph, which assumes the actual means are presented somewhere else in the paper (such as a table).

We performed a mixed-model analysis of variance, treating *method* as a fixed effect and *participant* as a random effect. The omnibus test showed a significant main effect of *method* (*F*(____ , ____) = ____, *p* < ____), prompting us to investigate pairwise differences. We employed Tukey's HSD procedure to address the increased risk of Type I error due to unplanned comparisons, finding that *speech* leads to significantly greater *WPM* than both *keyboard* (*F*(____ , ____) = ____, *p* < ____) and *Graffiti* (*F*(____ , ____) = ____, *p* < ____) and that *keyboard* also leads to significantly greater *WPM* than *Graffiti* (*F*(____ , ____) = ____, *p* < ____).

Study 2: Comparison of Multiple Interfaces

This data was published in the following UIST 2007 paper:

> Raphael Hoffmann, James Fogarty, Daniel S. Weld. (2007). Assieme: Finding and Leveraging Implicit References in a Web Search Interface for Programmers. *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST 2007)*. pp. 13-22.

This is a real dataset collected in a within-subjects experiment comparing participant completion of ten tasks in each of three interfaces. The completion of a task in any interface rendered it useless for testing the other interfaces (knowing the answer from completing a task once rendered that task a poor measure of the other interfaces). The experimenters therefore assembled a library of forty tasks, presented ten random tasks to participants during an initial practice stage, and then presented ten random tasks for each of the three interfaces (each participant therefore completed each of the forty tasks exactly once). Interfaces were presented using a counterbalanced design.

The data file contains eleven columns:

Independent Variables

Participant: A unique identifier for the participant

Trial: Which trial this is for the participant (range 1 to 30)

Interface: Which interface the participant is using (A, B, C)

Task: A unique identifier for the task

TaskSel: Whether or not the task has the Sel property, an important type of task (0 or 1)

TaskCoo: Whether or not the task has the Coo property, an important type of task (0 or 1)

TaskUse: Whether or not the task has the Use property, an important type of task (0 or 1)

TaskEx: Whether or not the task has the Ex property, an important type of task (0 or 1)

Dependent Variables

Restarts: A count of the number times a participant chose to restart the task

Time: Total time spent on the task

Correctness: An expert rating of the quality of the participant's solution to the task

You will find the instructions provided here are much less detailed than those for Study 1, but hopefully still contain enough to guide you on a correct path.

2.1 What type did you assign to each of the variables. Briefly, why?

Analyzing Time

Run a mixed-model analysis of variance for *Time*, as estimated by the independent variables.

2.2 Insert a screenshot of your configuration of the Run Model Dialog.

2.3 Insert a screenshot of the resulting output.

You will notice that a number of the independent variables above have no significant impact on *Time*. It is common to therefore remove them from the remainder of your analysis of *Time*. Run a new mixed-model analysis of variance for *Time*, using only the variables that were significant.

2.4 Insert a screenshot of your configuration of the Run Model Dialog.

2.5 Insert a screenshot of the resulting output.

Trial has a significant effect (if not, check your work up to this point), but is not our variable of interest. The negative parameter estimate means that people took less *Time* to complete the task as *Trial* increased (maybe they got better at the task, maybe they got sick of it and did not try as hard). What we should be concerned about is whether *Trial* affects *Interface*. Add an additional parameter to the model, created by crossing *Trial* with *Interface*. This is an interaction.

2.6 Insert a screenshot of your configuration of the Run Model Dialog.

2.7 Insert a screenshot of the resulting output.

2.8 Does *Trial* significantly interact with *Interface*? What does this mean?

Go back to working with the same variables you had in 2.4. *Interface* has a significant effect, so test significance of the difference between the levels of *Interface* and obtain the pairwise contrasts.

2.9 Insert a screenshot of your final dialog analyzing *Time*.

Once again, you have now captured everything you need to report an analysis of *Time*.

2.10 Prepare a description of your analysis that resembles that in 1.6.

Note that your analysis here is more complex and the resulting description will be longer. After all, you removed variables that were not significant, checked an interaction, interpreted that interaction, and only then conducted your analysis of the *Interface* variable.

Analyzing Restarts

Run a mixed-model analysis of variance for *Restarts*, as estimated by the independent variables.

2.11 Prepare a description of your analysis that resembles that in 1.6.

Here you are asked to perform the entire analysis on your own. Only the final description is strictly necessary, but it will be easier to award partial credit if you show your work.

Analyzing Correctness

Run a mixed-model analysis of variance for *Correctness*, as estimated by the independent variables.

2.12 Prepare a description of your analysis that resembles that in 1.6.

Again you are asked to perform the entire analysis on your own. Again only the final description is strictly necessary, but it will be easier to award partial credit if you show your work.

Summary

You have now analyzed three different independent measures each intended to give some insight into the appropriateness of the three different interfaces studied here. Summarize your results and indicate which interface seems to be the best for these tasks.

2.13 Summarize your overall results.

Study 3: Comparison of Multiple Conditions

This data was published in this CHI 2016 paper:

> Xiaoyi Zhang, Laura R. Pina, James Fogarty. (2016). Examining Unlock Journaling with Diaries and Reminders for In Situ Self-Report in Health and Wellness. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2016).*

This is a real dataset collected in a within-subjects experiment comparing participant data logging with six conditions (two interfaces, three notification conditions). Participants selected the type of data they were most interested in journaling (sleepiness, pleasure and accomplishment, or mood). Participants used an Android application over 18 days. Each condition was used for two consecutive days, with a rest day between conditions. Conditions were presented using a counterbalanced design.

The two interfaces consisted of an app which participants could open and log data from (*without unlock*), and a lock-screen interaction for logging data (*with unlock*). Participants either received no notifications (*none*), *traditional* notifications consisting of a push notification, or *aggressive* notifications which added sound and vibration to the push notification. Participants could journal at any time from within the app in all conditions. This was the only option in the *none*, *without unlock* condition. Participants were asked to journal ever 30 minutes for a 12-hour window. Notifications were delivered every 30 minutes in the two notification conditions.

The data file contains eleven columns:

Independent Variables

Participant: A unique identifier for the participant

Gender: The gender the participant identified as

Type: The type of data the participant entered (S = Sleepiness, P = Pleasure, M = Mood)

CalendarDay: Calendar day since the beginning of the study

StudyDay: Day since the beginning of the study only considering logging days

WithUnlock: Whether or not the participant had the lock-screen interaction that day (0 or 1)

Notification: Notifications received that day (N = None, T = Traditional, A = Aggressive)

Dependent Variables

Intrusiveness: Likert rating for interaction intrusiveness that day (1 to 5)

Frequency: Number of times participant logged data that day

Timeliness: Average number of minutes to nearest journaling interval (-15 to 15)

These instructions are even less detailed than the previous two studies, but more closely approximate a real-world analysis.

Analysis

Analyze the data and find what contributes to the dependent variables. It is up to you to determine the types of independent variables and select an appropriate model to fit. You must show us that you analyzed this data, but do not need to describe your analysis in detail. One way of doing this would be to attach your JMP notebook or screenshots from your results, with a few comments explaining why you did certain things (see the Study 1 screenshots for an example).

3.1 Show us how you analyzed this data.

Summary

You have now analyzed three different independent measures which provide some insight into the six study conditions. Summarize your results as you might see in a published paper. Of course, you could read simplify the final paper and see how the authors summarized their analysis. But treat this as an exercise in learning to describe the results yourself.

3.2 Summarize your overall results.