STAT 601 Final Project

Yuchi Hu

November 18, 2018

Abstract

Keystroke dynamics is the analysis of the timing, or rhythm, of an individual's typing on a computer keyboard. Detailed timing information on when each key is pressed and released is collected to describe an individual's typing rhythm, which can be thought of as a digital fingerprint. Similar to how physical fingerprints at a crime scene can be used to uniquely identify suspects, typing rhythms can be used to distinguish between a genuine user and an impostor.

Our objective in this project is to investigate how a person's typing dynamics changes over time. We will be building and comparing different models based on the data set collected by Kevin Killourhy and Roy Maxion of Carnegie Mellon University [1].

1. Problem Statement

Killourhy and Maxion [1] collected a data set and developed an evaluation procedure to measure and compare the performances of 14 anomaly-detection algorithms for keystroke dynamics. The data set is comprised of various timing features such as time between key presses and time each key is held down. We will analyze this data set to investigate how a person's typing dynamics changes over time. For example, does a person's typing speed increase over time? How do capital letters, numbers, and special characters affect a person's typing dynamics? And so forth.

2. Background

The data was collected from 51 subjects from Carnegie Mellon University [1]. Each subject completed 8 data-collection sessions with at least one day between sessions. During a data-collection session, subjects are required to correctly type a password 50 times. The password chosen is **.tie5Roanl**, which is representative of a strong 10-character password. When the subject presses or releases a key, the keyboard records the event as keydown or keyup, the name of the key, and the time the event occurred.

The first three variables of this data set are *subject* (for the 51 subjects), *sessionIndex* (for the 8 sessions), and *rep* (for the 50 times the password is entered for each session). The rest of the variables contain timing information for keydown-keydown times, keyup-keydown times, and hold times for the keys in the password. For example, *H.period* is the amount of time the "." key is held down; *DD.period.t* is the amount of time between the presses of the "." and "t" keys; and *UD.period.t* is the amount of time between the release of the "." key and the press of the "t" key.

Since we are interested in how a person's typing dynamics changes over time, the total time it takes to type the password will be the response variable, which can be obtained indirectly from the data set by summing the "DD" (keydown-keydown) variables and the final "H" (hold) variable (*H.Return*). Note that we are not including the "UD" (keyup-keydown) variables in the sum due to time overlap.

3. Exploratory Analysis

3.1 Timing Variables (Keydown-keydown, Keyup-keydown, and Hold)

First, we examine the box plots of the timing variables on separate plots, grouped by name (“DD”, “UD”, and “H”).

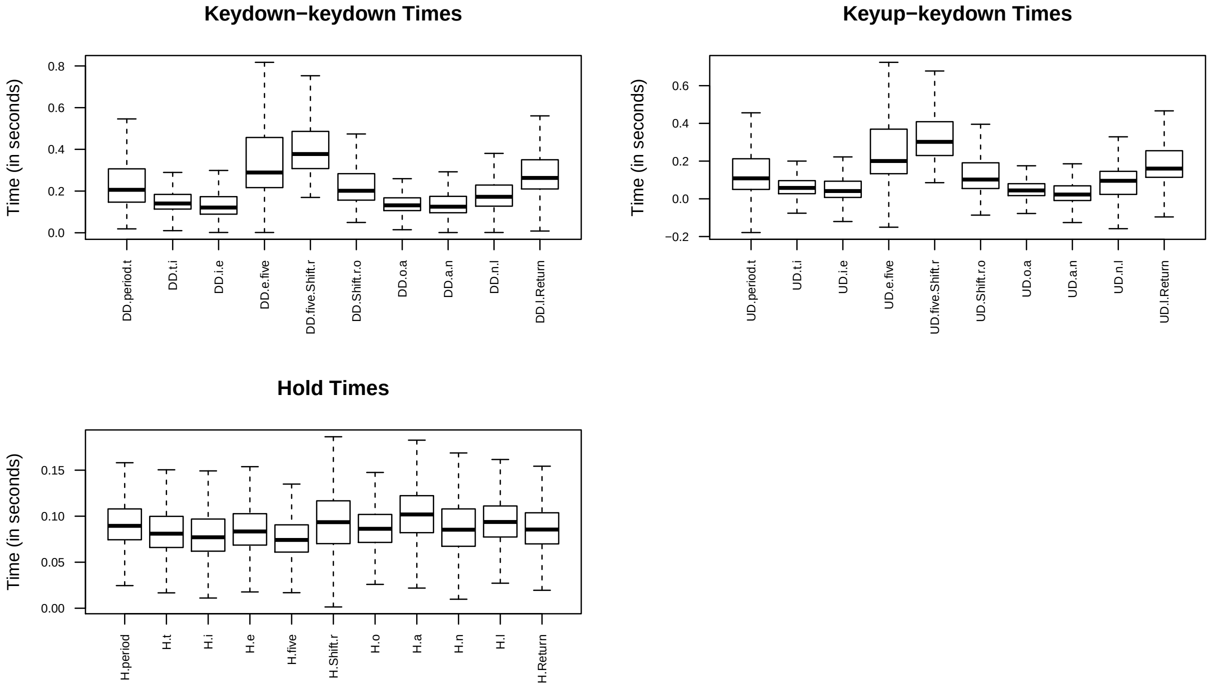

Figure 1: Box plots of keydown-keydown, keyup-keydown, and hold times.

We can see from **Figure 1** that keydown-keydown (“DD”) and keyup-keydown (“UD”) times follow essentially the same pattern; however, “DD” times are longer than their corresponding “UD” times. The similar patterns of the “DD” and “UD” plots indicate that the two groups of variables are highly correlated and thus redundant. If we look at the patterns more closely, we can see that typing two lowercase letters sequentially takes shorter amounts of time (the variabilities in those times are also smaller) than other combinations of characters. This makes sense since typing “R” (capital r) requires pressing an extra key (“Shift”) and pressing the “.”, “5”, and “Return” keys are more difficult due to finger placement.

As for hold (“H”) times, we would expect all of the keys to be held down for about the same amount of time. But interestingly, the “a” key appears to be held down the longest. It is not so obvious why this is the case.

Note that outliers are not displayed to improve visualization.

3.2 Total Time (Response Variable)

Next, we examine histograms of the response variable *total.time*, which represents the total time it takes to type the password and is obtained by summing the "DD" variables and the final "H" variable (*H.Return*).

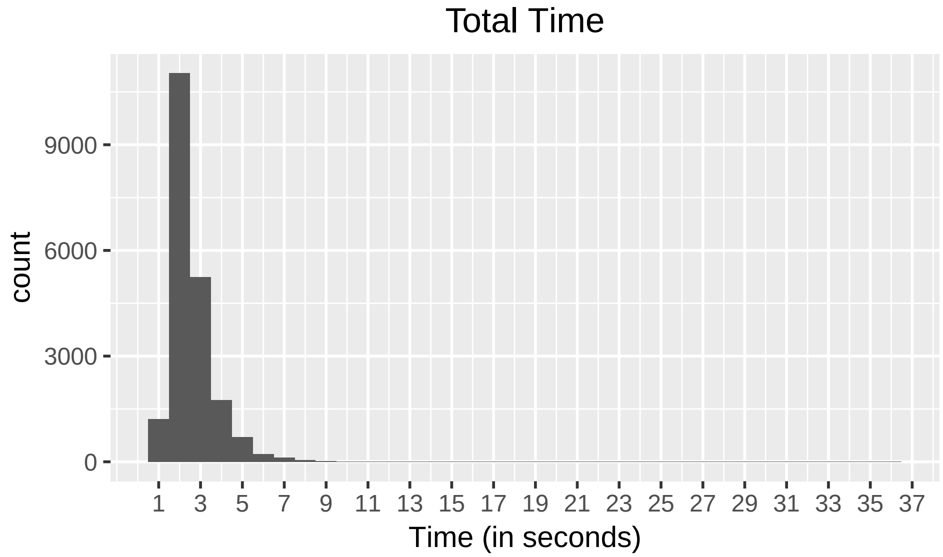

Figure 2: Histogram of total time.

We can see from **Figure 2** that *total.time* follows a right-skewed distribution and that the total time it takes to type the password is about two seconds in most of the cases. Note that the plot has quite a bit of empty space which is caused by the unusual *total.time* value of 36 seconds. We will assume that unusual values are simply due to slow typing.

We might also be interested in how *total.time* changes with each session since we would naturally assume a subject to become more proficient in typing the password over time.

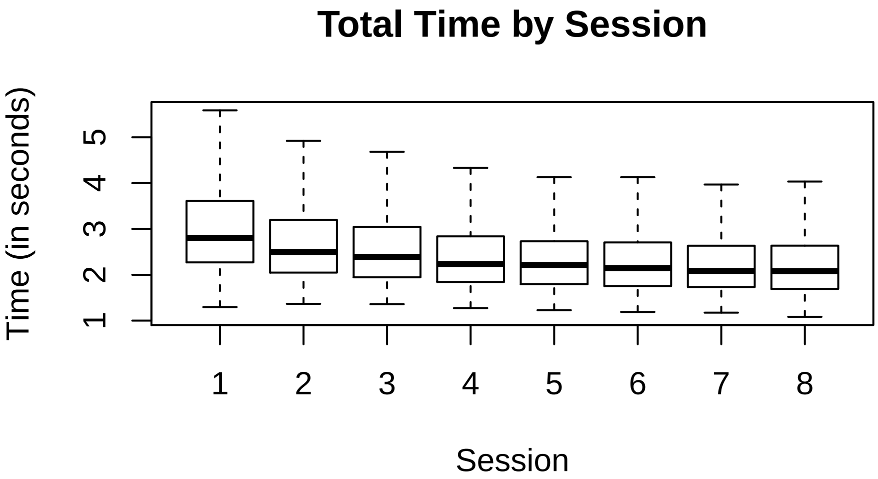

Figure 3: Box plots of total time by session.

We can see from **Figure 3** that *total.time* decreases from session 1 to session 4 then stabilizes thereafter, which is indicative of the subjects becoming more and more proficient until no further improvements were possible.

Similarly, we would expect subjects to become more proficient as they progress through a session; that is, we would expect *total.time* to decrease as *rep* increases within a session.

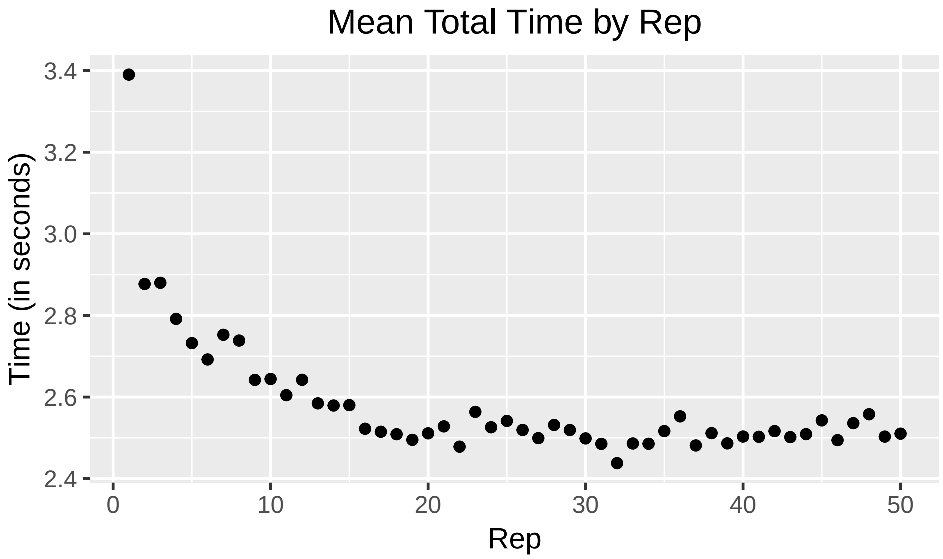

Figure 4: Scatter plot of mean total time by rep.

We can see from **Figure 4** that there's a relatively significant decrease in *total.time* from rep 1 to rep 2. From rep 2, *total.time* steadily decreases until about rep 15; from there, no further improvements in proficiency can be discerned.

3.3 Summary

From the exploratory analysis, we found that typing two lowercase letters sequentially takes shorter amounts of time than other combinations of characters. We also found that typing proficiency improves from session 1 to session 4 as well as from rep 1 to rep 15 within a session.

4. Models

4.1 Multiple Linear Regression

We fit a multiple linear regression model with *total.time* as the response and *sessionIndex*, *rep*, and the interaction of *sessionIndex* and *rep* as the predictors. The summary of the model is below:

```
##
## Call:
## glm(formula = total.time ~ sessionIndex * rep, data = x)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.812  -0.685  -0.260   0.353  32.730
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.5607557  0.0343633 103.621   <2e-16 ***
## sessionIndex     -0.1793644  0.0068050 -26.358   <2e-16 ***
## rep              -0.0163540  0.0011728 -13.944   <2e-16 ***
## sessionIndex:rep  0.0021232  0.0002322   9.142   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.203052)
##
##     Null deviance: 26514  on 20399  degrees of freedom
## Residual deviance: 24537  on 20396  degrees of freedom
## AIC: 61670
##
## Number of Fisher Scoring iterations: 2
```

We can see that both predictors as well as the interaction term are highly significant at an alpha level of 0.05. The AIC of the model is 61670.

4.2 Regression Tree

We fit an rpart (recursive partitioning and regression trees) model using the **rpart** function from the **rpart** package with *total.time* as the response and *sessionIndex* and *rep* as the predictors. We plot the regression tree using the **ggdendro** package.

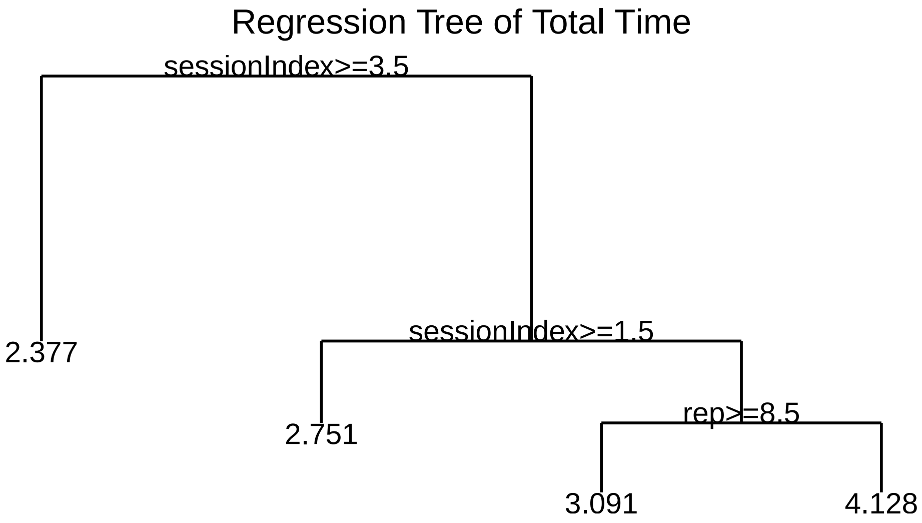

Figure 5: Regression tree of total time.

We can see from **Figure 5** that *total.time* is lower for larger values of *sessionIndex* and for larger values of *rep*, which indicates increased typing proficiency in later sessions and later reps within a session.

4.3 Quantile Regression

We plot *total.time* vs. *sessionIndex* and *total.time* vs. *rep* separately, overlaid with linear quantile regression lines for the 25% (red), 50% (blue), and 75% (green) quantiles. Note that the data points are omitted from the plots due to clutter.

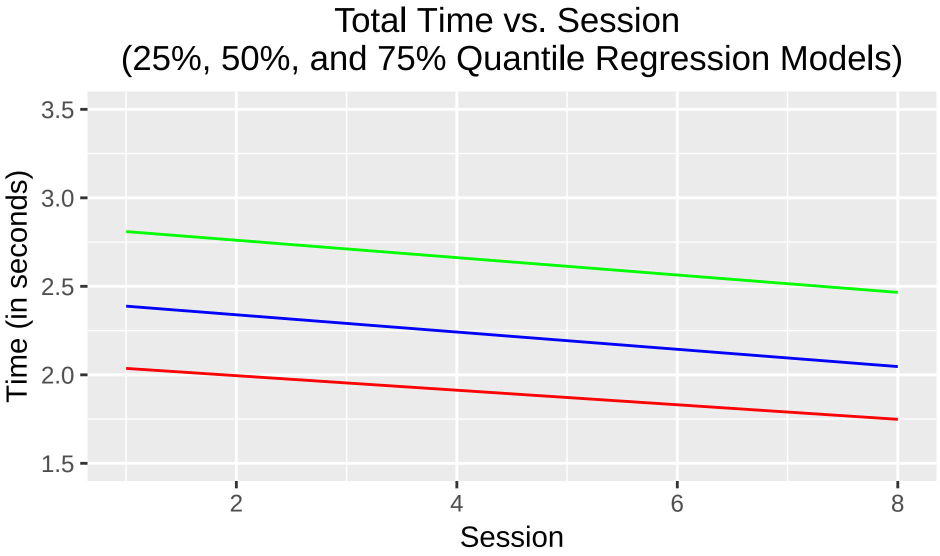

Figure 6: 25%, 50%, and 75% quantile regression models for total time vs. session.

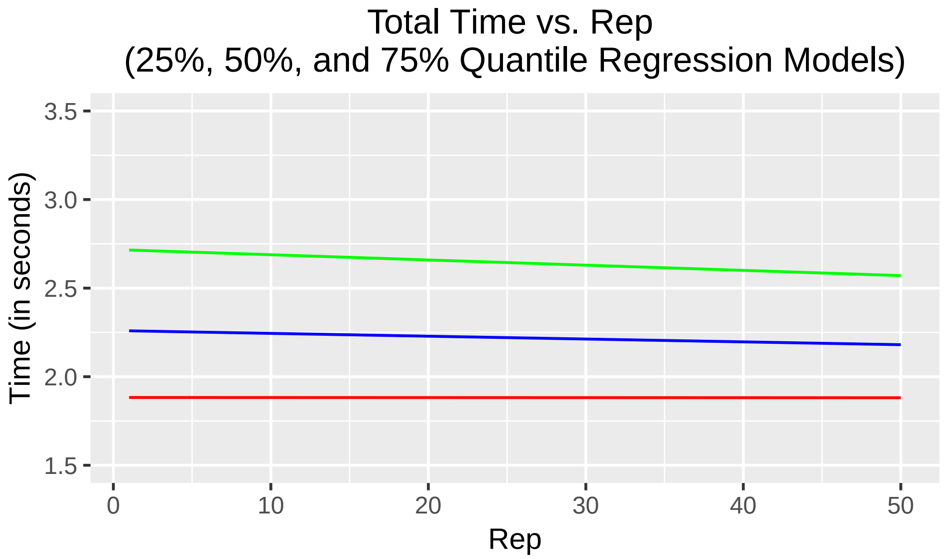

Figure 7: 25%, 50%, and 75% quantile regression models for total time vs. rep.

We can see from **Figure 6** that the slopes of the quantile regression lines are approximately equal and decreasing with each subsequent session. On the other hand, we can see from **Figure 7** that only the slope of the 75% quantile regression line is noticeably decreasing with each subsequent rep.

In other words, with each subsequent session, subjects of all typing speeds are becoming more proficient. Whereas, with each subsequent rep, the slow typers (in the 75% quantile of *total.time*) are becoming more proficient, but the typing speeds of the faster typers (in the 25% and 50% quantile of *total.time*) are barely affected, if at all, with more reps.

Table 1: AIC Comparison of Quantile Regression Models

25% Quantile	49081.53
Median	54795.95
75% Quantile	66309.88

Table 2 shows the AIC's of the quantile regression models. The increasing trend of the AIC's indicates that *total.time* is more variable in the upper quantiles.

4.4 Linear Mixed-Effects

We fit a random intercept model and a random intercept and slope model using the **lmer** function from the **lme4** package with *sessionIndex*, *rep*, and the interaction between *sessionIndex* and *rep* as fixed effect covariates. For the random intercept model, *subject* is included as a random effect, while for the random intercept and slope model, *sessionIndex* and *subject* are included as random effects (random slope and random intercept, respectively). Random effects identify the source of the repeated measurements.

The summary of the random intercept model is below:

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: total.time ~ sessionIndex * rep + (1 | subject)
##    Data: x
## 
##      AIC      BIC   logLik deviance df.resid 
##  42656.5  42704.0 -21322.2  42644.5    20394 
## 
## Scaled residuals: 
##    Min     1Q Median     3Q    Max 
## -3.231 -0.470 -0.117  0.286 44.292 
## 
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  subject  (Intercept) 0.7368   0.8584  
##  Residual             0.4660   0.6826  
## Number of obs: 20400, groups:  subject, 51
## 
## Fixed effects:
##                    Estimate Std. Error t value
## (Intercept)       3.5607557  0.1220858   29.17
## sessionIndex     -0.1793644  0.0042352  -42.35
## rep              -0.0163540  0.0007299  -22.41
## sessionIndex:rep  0.0021232  0.0001445   14.69
## 
## Correlation of Fixed Effects:
##             (Intr) sssnIn rep   
## sessionIndx -0.156              
## rep         -0.152  0.776       
## sssnIndx:rp  0.136 -0.870 -0.891
```

And the summary of the random intercept and slope model is below:

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: total.time ~ sessionIndex * rep + (sessionIndex | subject)
##    Data: x
## 
##      AIC      BIC   logLik deviance df.resid 
##  40078.8  40142.2 -20031.4  40062.8    20392 
## 
## Scaled residuals: 
##    Min     1Q Median     3Q    Max 
## -4.990 -0.472 -0.131  0.277 44.114 
## 
## Random effects:
```

```
##  Groups   Name         Variance Std.Dev. Corr
##  subject  (Intercept)  1.5601   1.2490
##           sessionIndex 0.0112   0.1058   -0.88
##  Residual              0.4070   0.6380
## Number of obs: 20400, groups:  subject, 51
##
## Fixed effects:
##                   Estimate Std. Error t value
## (Intercept)       3.5607557  0.1760393   20.23
## sessionIndex     -0.1793644  0.0153380  -11.69
## rep              -0.0163540  0.0006822  -23.97
## sessionIndex:rep  0.0021232  0.0001351   15.72
##
## Correlation of Fixed Effects:
##             (Intr) sssnIn rep
## sessionIndx -0.873
## rep         -0.099  0.200
## sssnIndx:rp  0.088 -0.225 -0.891
```

Table 2: AIC Comparison of Linear Mixed-Effects Models

Random Intercept	42656.49
Random Intercept and Slope	40078.77

Table 2 shows the AIC's of the two linear mixed-effects models. We can see that the random intercept and slope model has a lower AIC and thus is a better fit for the data.

Additionally, a random slope and intercept model might be more sensible since it allows each individual's regression line to differ in intercept and in slope from the regression lines of other individuals. This is in contrast to a random intercept model, in which an individual's regression line can differ in intercept but not in slope from the regression lines of other individuals. Furthermore, a random intercept model constrains the variance of each repeated measure to be the same and the covariance between any pair of measurements to be equal. These constraints are often not realistic for repeated measures data [2]. Thus, a random slope and intercept model is more appropriate since it allows a more realistic structure for the covariances.

4.5 Generalized Estimating Equations

We fit two GEE models with independence and exchangeable correlation structures. The summaries of the two GEE models are below:

Table 3: Summary for the x.gee1 Model (Correlation Structure = Independence)

	Estimate	Naive S.E.	Naive z	Robust S.E.	Robust z
(Intercept)	3.5607557	0.0313295	113.65516	0.2024283	17.59021
sessionIndex	-0.1793644	0.0062042	-28.91035	0.0173916	-10.31329
rep	-0.0163540	0.0010693	-15.29473	0.0014714	-11.11445
sessionIndex:rep	0.0021232	0.0002117	10.02739	0.0001907	11.13657

Table 4: Summary for the x.gee2 Model (Correlation Structure = Exchangeable)

	Estimate	Naive S.E.	Naive z	Robust S.E.	Robust z
(Intercept)	3.5607557	0.1113080	31.99011	0.2024283	17.59021
sessionIndex	-0.1793644	0.0038622	-46.44052	0.0173916	-10.31329
rep	-0.0163540	0.0006656	-24.56890	0.0014714	-11.11445
sessionIndex:rep	0.0021232	0.0001318	16.10764	0.0001907	11.13657

We can see from **Table 4** that the exchangeable correlation model has naive and robust standard errors that are closer to each other, indicating that it is a better model than the independence correlation model. We can also compare their QIC's using the **QIC** function from the **MESS** package. The outputs of the **QIC** function for each model are summarized in the table below:

Table 5: QIC Comparison of GEE Models

	Independence	Exchangeable
QIC	25075.6681	24661.75179
QICu	24545.4418	24545.44176
Quasi Lik	-12268.7209	-12268.72088
CIC	269.1132	62.15502
params	4.0000	4.00000
QICC	25075.6701	24661.75473

We can see from **Table 5** that the exchangeable correlation model has the lower QIC, indicating again that it is better than the independence correlation model.

4.6 Multiple Comparisons

We perform a multiplicity adjusted test on all regression coefficients (except for the intercept) being zero. We set up a matrix K, which reads:

```
##                  [,1] [,2] [,3] [,4]
## sessionIndex        0    1    0    0
## rep                 0    0    1    0
## sessionIndex:rep    0    0    0    1
```

The **glht** function takes the fitted linear regression model and a description of the matrix K to perform a multiplicity adjusted test. The summary of the **glht** object is below:

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Fit: glm(formula = total.time ~ sessionIndex * rep, data = x)
##
## Linear Hypotheses:
##                         Estimate Std. Error z value Pr(>|z|)
## sessionIndex == 0     -0.1793644  0.0068050 -26.358   <2e-16 ***
## rep == 0              -0.0163540  0.0011728 -13.944   <2e-16 ***
## sessionIndex:rep == 0  0.0021232  0.0002322   9.142   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

We can see that all of the regression coefficients are still significant at an alpha level of 0.05.

We may also be interested in the pairwise differences between the mean *total.time* of each session.

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = total.time ~ as.factor(sessionIndex), data = x)
##
## $`as.factor(sessionIndex)`
##             diff         lwr          upr     p adj
## 2-1 -0.4441045882 -0.53723861 -0.350970568 0.0000000
## 3-1 -0.5677554510 -0.66088947 -0.474621431 0.0000000
## 4-1 -0.7328619608 -0.82599598 -0.639727941 0.0000000
## 5-1 -0.8223253725 -0.91545939 -0.729191353 0.0000000
## 6-1 -0.9050814118 -0.99821543 -0.811947392 0.0000000
## 7-1 -0.9693330980 -1.06246712 -0.876199078 0.0000000
## 8-1 -0.9701466667 -1.06328069 -0.877012647 0.0000000
## 3-2 -0.1236508627 -0.21678488 -0.030516843 0.0014792
## 4-2 -0.2887573725 -0.38189139 -0.195623353 0.0000000
## 5-2 -0.3782207843 -0.47135480 -0.285086764 0.0000000
## 6-2 -0.4609768235 -0.55411084 -0.367842804 0.0000000
## 7-2 -0.5252285098 -0.61836253 -0.432094490 0.0000000
## 8-2 -0.5260420784 -0.61917610 -0.432908058 0.0000000
## 4-3 -0.1651065098 -0.25824053 -0.071972490 0.0000021
## 5-3 -0.2545699216 -0.34770394 -0.161435902 0.0000000
## 6-3 -0.3373259608 -0.43045998 -0.244191941 0.0000000
## 7-3 -0.4015776471 -0.49471167 -0.308443627 0.0000000
## 8-3 -0.4023912157 -0.49552524 -0.309257196 0.0000000
## 5-4 -0.0894634118 -0.18259743  0.003670608 0.0702803
## 6-4 -0.1722194510 -0.26535347 -0.079085431 0.0000005
## 7-4 -0.2364711373 -0.32960516 -0.143337117 0.0000000
## 8-4 -0.2372847059 -0.33041873 -0.144150686 0.0000000
## 6-5 -0.0827560392 -0.17589006  0.010377981 0.1244985
## 7-5 -0.1470077255 -0.24014175 -0.053873706 0.0000471
## 8-5 -0.1478212941 -0.24095531 -0.054687274 0.0000413
## 7-6 -0.0642516863 -0.15738571  0.028882334 0.4206009
## 8-6 -0.0650652549 -0.15819927  0.028068765 0.4033465
## 8-7 -0.0008135686 -0.09394759  0.092320451 1.0000000
```

We can see that most of the pairwise differences between sessions are significant except the ones between sessions 4 & 5, 5 & 6, 6 & 7, 6 & 8, and 7 & 8. This is an indication that typing proficiency significantly improves from session to session until session 4 but only marginally improves thereafter.

In the case of multiple testing, we are testing whether the coefficients are significant simultaneously, and the resulting p-values are adjusted for the family-wise error rate. Essentially, multiple testing procedures take the correlation among the estimated coefficients of interest into account when computing p-values [2]. The p-values obtained from the multiplicity adjusted test are computed from data of ALL the comparisons, so the p-value of a specific comparison would change if the data of the other comparisons or the number of comparisons is changed.

5. Conclusion

From the multiple linear regression model, we found that both *sessionIndex* and *rep* as well as their interaction are highly significant at an alpha level of 0.05. From the regression tree, we saw that there are three splits: *sessionIndex* $\geq$ 3.5, *sessionIndex* $\geq$ 1.5, and *rep* $\geq$ 8.5. In other words, typing proficiency tends to be higher (lower *total.time*) at later sessions and later reps within a session. From the quantile regression models, we found that subjects of all typing speeds became more proficient with each subsequent session, while only the slow typers (in the 75% quantile of *total.time*) became more proficient with each subsequent rep.

In the previous models, we assumed that the data is independent. However, since each subject typed the password 50 times in a session as well as participated in eight different sessions, we cannot make that assumption. Thus, we included *subject* and *sessionIndex* + *subject* as random effects in the random intercept and random intercept and slope models (linear mixed-effects models), respectively. We found that the random intercept and slope model is a better fit for the data. For the generalized estimating equations, we found that the model with the exchangeable correlation structure is better than that with the independence correlation structure.

The question of which model is the best is rather open-ended. We have found interesting patterns in *total.time* from the regression tree and quantile regression plots; however, they assume that the data is independent. Since this study is longitudinal, a mixed-effects model or a generalized estimating equation approach is preferable. GEE's model the **population average** effect of the covariates, while mixed-effects models model the **subject-specific** effect of the covariates. Which one we choose will depend on whether we are interested in the population average effect or subject-specific effect of the covariates.

References

[1] Killourhy, K. & Maxion, R. (2009). Comparing Anomaly Detectors for Keystroke Dynamics. *39th Annual International Conference on Dependable Systems & Networks*, 125-134.

[2] Everitt, B. & Hothorn, T. (2014). *A Handbook of Statistical Analyses using R, Third Edition.* Boca Raton, FL: CRC Press.