Semantic Networks Analysis

Abhilasha Kumar

January 16, 2018

Reading the Data

```
library(dplyr)
sem = read.csv("NetworksProject_AllSubjects.csv", header = TRUE, sep = ",")
sem$mean_len = as.double(as.character(sem$mean_len))
sem$mean_logf = as.double(as.character(sem$mean_logf))
sem$mean_ldtz = as.double(as.character(sem$mean_ldtz))
sem$mean_nmgz = as.double(as.character(sem$mean_nmgz))
sem$undirected = as.double(as.character(sem$undirected))
sem$directed = as.double(as.character(sem$directed))
sem$logp = as.double(as.character(sem$logp))

sem$mean_len = round(sem$mean_len, 2)
sem$mean_logf = round(sem$mean_logf, 2)
sem$mean_ldtz = round(sem$mean_ldtz, 2)
sem$mean_nmgz = round(sem$mean_nmgz, 2)

itemlist = read.csv("KenettCompleteItemList.csv", header = TRUE, sep = ",")
colnames(itemlist) = c("prime_word", "target_word", "ItemNo", "List", "proc")
itemlist$prime_word = paste(itemlist$prime_word, "")
itemlist$target_word = paste(itemlist$target_word, "")
sem$prime_word = as.character(sem$prime_word)
sem$target_word = as.character(sem$target_word)
sem$proc = as.character(sem$proc)
itemlist$proc = as.character(itemlist$proc)

## Problem in ITEM NAIVE-INTEREST due to coding issues
## So we replae this item by simple text

sem$prime_word = ifelse(sem$prime_word == "NAÃ\u008fVE ", "NA\xecVE ",
                        sem$prime_word)
sem$target_word = ifelse(sem$target_word == "NAÃ\u008fVE ", "NA\xecVE ",
                        sem$target_word)

sem = inner_join(sem, itemlist, by = c("prime_word",
                                          "target_word", "proc"))

sem = sem[order(sem$subject),]

kenettproc= group_by(sem, ItemNo) %>%
  summarise(items = n())
```

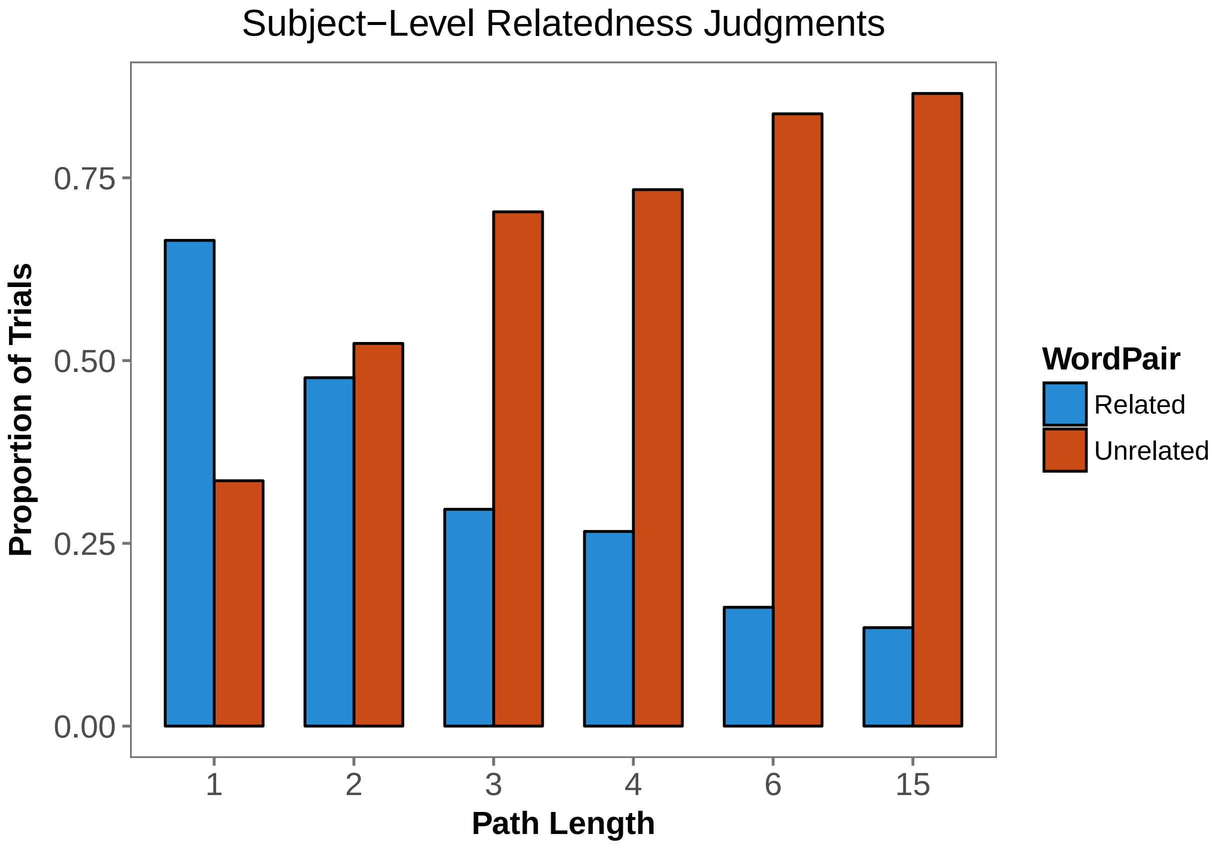
Related-Unrelated Decisions

```r
library(dplyr)

sem_decision = group_by(sem, subject, pathlength, Type ) %>%
  summarize(Trials = n())

sem_subjectrows  = group_by(sem_decision, subject) %>%
  summarise(numrows = n())
rows = which(sem_subjectrows$numrows < 12)

subjects = sem_subjectrows[rows,]

# With N = 80, there are 12 rowas missing!

sem_decision[949,] = c("42406", 15, "Related", 0)
sem_decision[950,] = c("61558", 15, "Related", 0)
sem_decision[951,] = c("27113", 15, "Related", 0)
sem_decision[952,] = c("34686", 6, "Related", 0)
sem_decision[953,] = c("81385", 6, "Related", 0)
sem_decision[954,] = c("35312", 3, "Related", 0)
sem_decision[955,] = c("65089", 1, "Unrelated", 0)
sem_decision[956,] = c("65448", 4, "Related", 0)
sem_decision[957,] = c("65448", 15, "Related", 0)
sem_decision[958,] = c("84694", 6, "Related", 0)

sem_decision[959,] = c("7416", 15, "Related", 0)
sem_decision[960,] = c("79426", 15, "Related", 0)



sem_decision$Trials = as.numeric(as.character(sem_decision$Trials))

sem_decision_agg = group_by(sem_decision, pathlength, Type ) %>%
  summarise_at(vars(Trials), mean)
```

Plotting Proportions

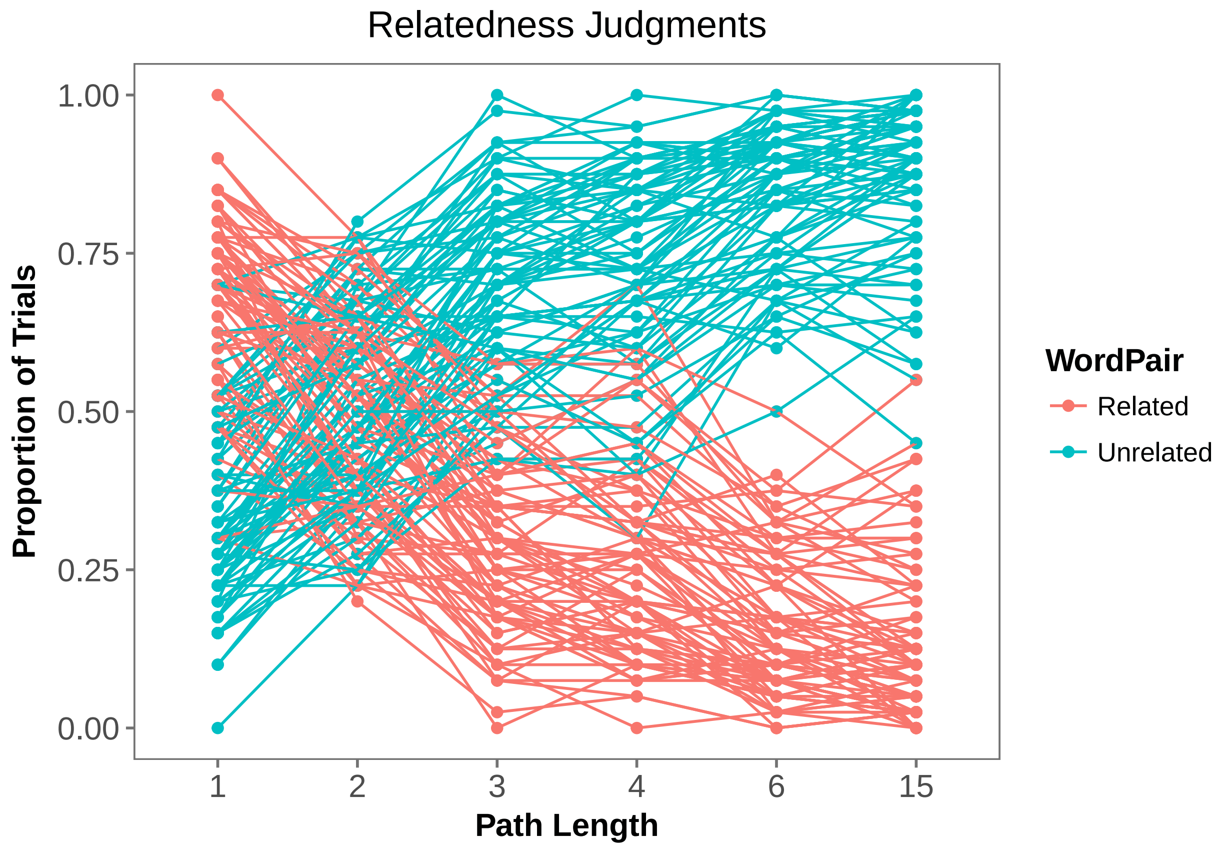
```r
sem_decision$Trials = as.numeric(as.character(sem_decision$Trials))
sem_decision_agg$Trials = as.numeric(as.character(sem_decision_agg$Trials))

sem_decision$proportion = sem_decision$Trials/40


sem_decision_agg$proportion = sem_decision_agg$Trials/(40)

sem_decision$pathlengthfac = ordered(as.factor(as.character(sem_decision$pathlength)),
                            levels = c("1", "2", "3", "4", "6", "15"))

sem_decision_agg$pathlengthfac = ordered(as.factor(as.character(sem_decision_agg$pathlength)),
                            levels = c("1", "2", "3", "4", "6","15"))
```

```r
library(ggplot2)
library(ggthemes)

sem_decision_agg = sem_decision_agg[order(sem_decision_agg$pathlength,
                                          sem_decision_agg$Type),]

sem_decision_agg %>% mutate(WordPair = factor(Type,
                                        levels = unique(Type),
                    labels = c("Related", "Unrelated")))%>%
  ggplot(aes(x = pathlengthfac, y = proportion,
                              fill = WordPair, group = WordPair))+
 geom_bar(stat = "identity", position = "dodge", width = 0.7, color= "black")+
 #geom_errorbar(aes(ymin=Trials - ci, ymax=Trials + ci),
  #           width=.2, color = "gray26",
   #          position = position_dodge(0.7))+
 theme_few()+
 scale_fill_solarized()+
    xlab("Path Length") + ylab("Proportion of Trials") +
  ggtitle("Subject-Level Relatedness Judgments")  +
   theme(axis.text = element_text(size = rel(1)),
          axis.title = element_text(face = "bold", size = rel(1)),
          legend.title = element_text(face = "bold", size = rel(1)),
         plot.title = element_text(hjust = .5),
         strip.text.x = element_text(face = "bold", size = rel(1.4)))
```



Line Plot Subject-Wise

```
sem_decision$subject = as.factor(as.character(sem_decision$subject))


sem_decision = sem_decision[order(sem_decision$subject,
                                  sem_decision$pathlength,
                                  sem_decision$Type),]

sem_decision %>% mutate(WordPair = factor(Type,
                                        levels = unique(Type),
                    labels = c("Related", "Unrelated")))%>%
  ggplot(aes(x = pathlengthfac, y = proportion,
             color = WordPair, group=interaction(WordPair, subject))) +
  geom_point()+
 geom_line() +
 theme_few()+
 scale_fill_solarized()+
 # facet_wrap(~subject)+
    xlab("Path Length") + ylab("Proportion of Trials") +
  ggtitle("Relatedness Judgments")  +
   theme(axis.text = element_text(size = rel(1)),
          axis.title = element_text(face = "bold", size = rel(1)),
          legend.title = element_text(face = "bold", size = rel(1)),
         plot.title = element_text(hjust = .5),
         strip.text.x = element_text(face = "bold", size = rel(1.4)))
```



ANOVA

```
relunrel_aov = aov(data = sem_decision, proportion ~ pathlengthfac*Type +
                     Error(subject/(pathlengthfac*Type)))
summary(relunrel_aov)
```

```
##
## Error: subject
##           Df    Sum Sq   Mean Sq F value Pr(>F)
## Residuals 79 1.308e-28 1.655e-30
##
## Error: subject:pathlengthfac
##                Df   Sum Sq   Mean Sq F value Pr(>F)
## pathlengthfac   5 4.60e-29 9.147e-30   0.767  0.574
## Residuals     395 4.71e-27 1.192e-29
##
## Error: subject:Type
##           Df Sum Sq Mean Sq F value Pr(>F)
## Type       1  26.62  26.617   176.7 <2e-16 ***
## Residuals 79  11.90   0.151
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: subject:pathlengthfac:Type
##                     Df Sum Sq Mean Sq F value Pr(>F)
## pathlengthfac:Type   5  32.74   6.547   412.2 <2e-16 ***
## Residuals          395   6.27   0.016
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
options(contrasts = c('contr.sum', 'contr.poly'))
 library(lsmeans)
```

```
## The 'lsmeans' package is being deprecated.
## Users are encouraged to switch to 'emmeans'.
## See help('transition') for more information, including how
## to convert 'lsmeans' objects and scripts to work with 'emmeans'.
```

```
 library(multcomp)
```

```
## Loading required package: mvtnorm

## Loading required package: survival

## Loading required package: TH.data

## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

##
## Attaching package: 'TH.data'

## The following object is masked from 'package:MASS':
```

```
##
##     geyser
```

```
 sem_lsm = lsmeans::lsmeans(relunrel_aov, c("pathlengthfac", "Type"))
```

```
## Warning in lsm.basis.aovlist(object, trms, xlev, grid, ...): Some predictors are correlated with the
## May help to re-fit with different contrasts, e.g. 'contr.sum'
```

```
 prime_effect = cld(sem_lsm, alpha = 0.05,
                 adjust = "tukey", details = TRUE, by = "pathlengthfac")
 library(knitr)
kable(subset(prime_effect$comparisons,prime_effect$comparisons$p.value < 0.05))
```

	contrast	pathlengthfac	estimate	SE	df	t.ratio	p.value
1	Related - Unrelated	1	0.328750	0.0309601	174.557	10.61852	0
3	Related - Unrelated	3	0.406875	0.0309601	174.557	13.14193	0
4	Related - Unrelated	4	0.467500	0.0309601	174.557	15.10010	0
5	Related - Unrelated	6	0.675000	0.0309601	174.557	21.80228	0
6	Related - Unrelated	15	0.730625	0.0309601	174.557	23.59895	0

Raw Reaction Time

```
sem_rt = group_by(sem, subject, pathlength ) %>%
  summarise_at(vars(rt), mean)

sem_rt_agg = group_by(sem, pathlength ) %>%
  summarise_at(vars(rt), mean)

sem_agg_rmisc = Rmisc::summarySE(sem_rt,
                        measurevar = "rt",
                        groupvars = c("pathlength"))
```

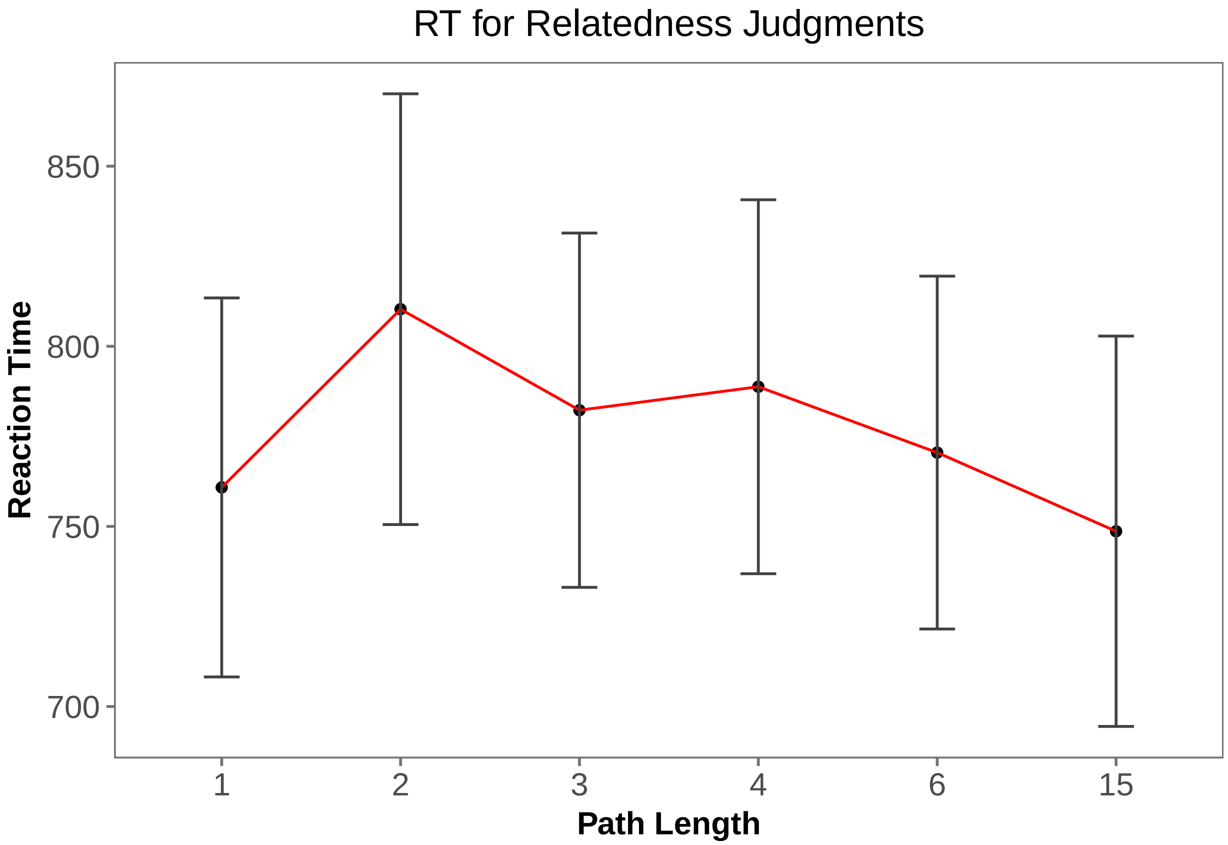
ANOVA

```
sem_rt$pathlengthfac = ordered(as.factor(as.character(sem_rt$pathlength)),
                            levels = c("1", "2", "3", "4", "6", "15"))
sem_rt$subject = as.factor(sem_rt$subject)
rt_aov = aov(data = sem_rt, rt ~ pathlengthfac +
                     Error(subject/(pathlengthfac)))
summary(rt_aov)
```

```
##
## Error: subject
##           Df   Sum Sq Mean Sq F value Pr(>F)
## Residuals 79 24339754  308098
##
## Error: subject:pathlengthfac
##                Df  Sum Sq Mean Sq F value   Pr(>F)
## pathlengthfac   5  190645   38129   6.167 1.61e-05 ***
## Residuals     395 2442052    6182
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

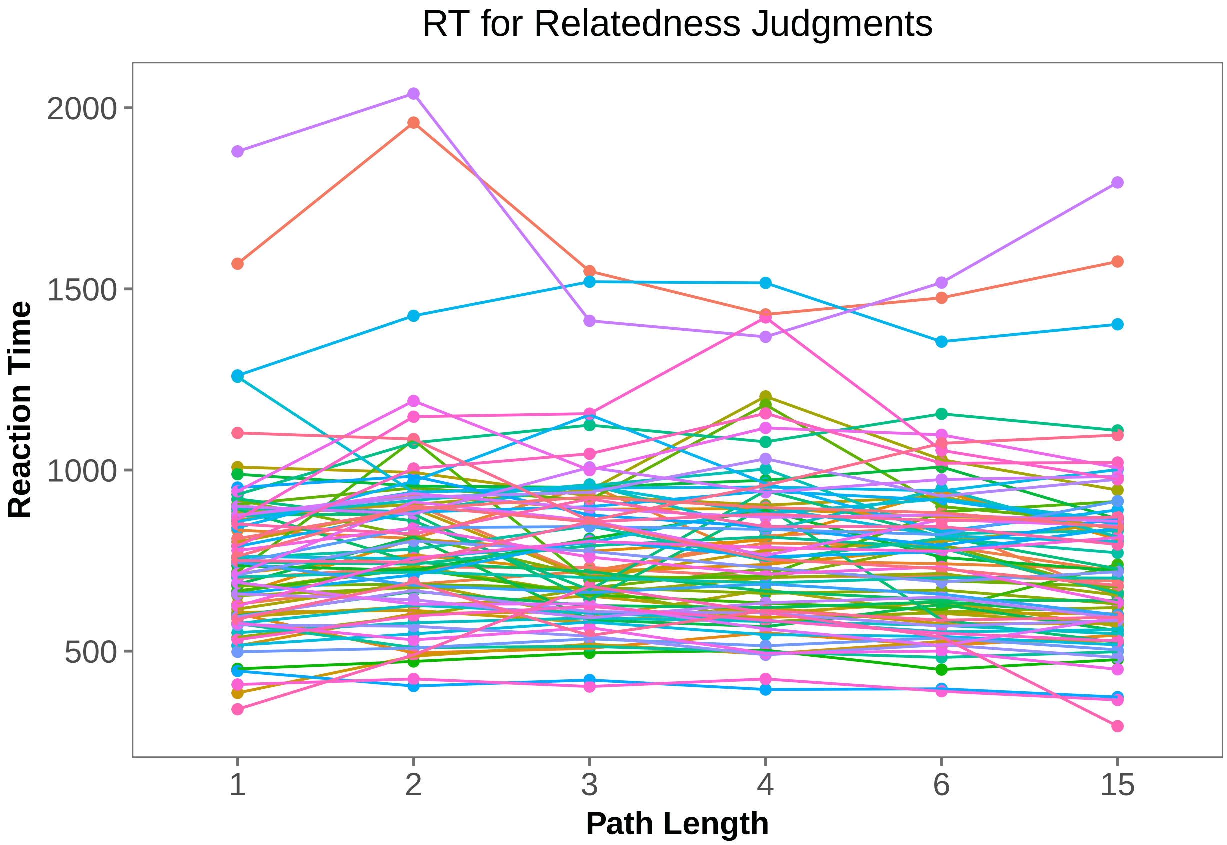
Plotting RTs

```
sem_agg_rmisc$pathlengthfac = ordered(as.factor(as.character(sem_agg_rmisc$pathlength)),

library(ggplot2)
library(ggthemes)

sem_agg_rmisc %>%
  ggplot(aes(x = pathlengthfac, y = rt, group = 1))+
  geom_point()+
  geom_line(color = "red")+
   geom_errorbar(aes(ymin=rt - ci, ymax=rt + ci),
             width=.2, color = "gray26",
             position = position_dodge(0.7))+
 theme_few()+
 # scale_x_continuous(breaks = c(1,2,3,4,5,6,10,15,20))+
    xlab("Path Length") + ylab("Reaction Time") +
  ggtitle("RT for Relatedness Judgments")  +
   theme(axis.text = element_text(size = rel(1)),
          axis.title = element_text(face = "bold", size = rel(1)),
          legend.title = element_text(face = "bold", size = rel(1)),
         plot.title = element_text(hjust = .5),
         strip.text.x = element_text(face = "bold", size = rel(1.4)))
```



Subject-Wise

```
library(ggplot2)
library(ggthemes)

sem_rt %>%
  ggplot(aes(x = pathlengthfac, y = rt,
             group = subject, color = subject))+
  geom_point()+
  geom_line()+
   #geom_errorbar(aes(ymin=Trials - ci, ymax=Trials + ci),
  #           width=.2, color = "gray26",
   #          position = position_dodge(0.7))+
 theme_few()+
  guides(color = FALSE)+
 # scale_x_continuous(breaks = c(1,2,3,4,6,15))+
    xlab("Path Length") + ylab("Reaction Time") +
  ggtitle("RT for Relatedness Judgments")  +
   theme(axis.text = element_text(size = rel(1)),
          axis.title = element_text(face = "bold", size = rel(1)),
          legend.title = element_blank(),
         plot.title = element_text(hjust = .5),
         strip.text.x = element_text(face = "bold", size = rel(1.4)))
```



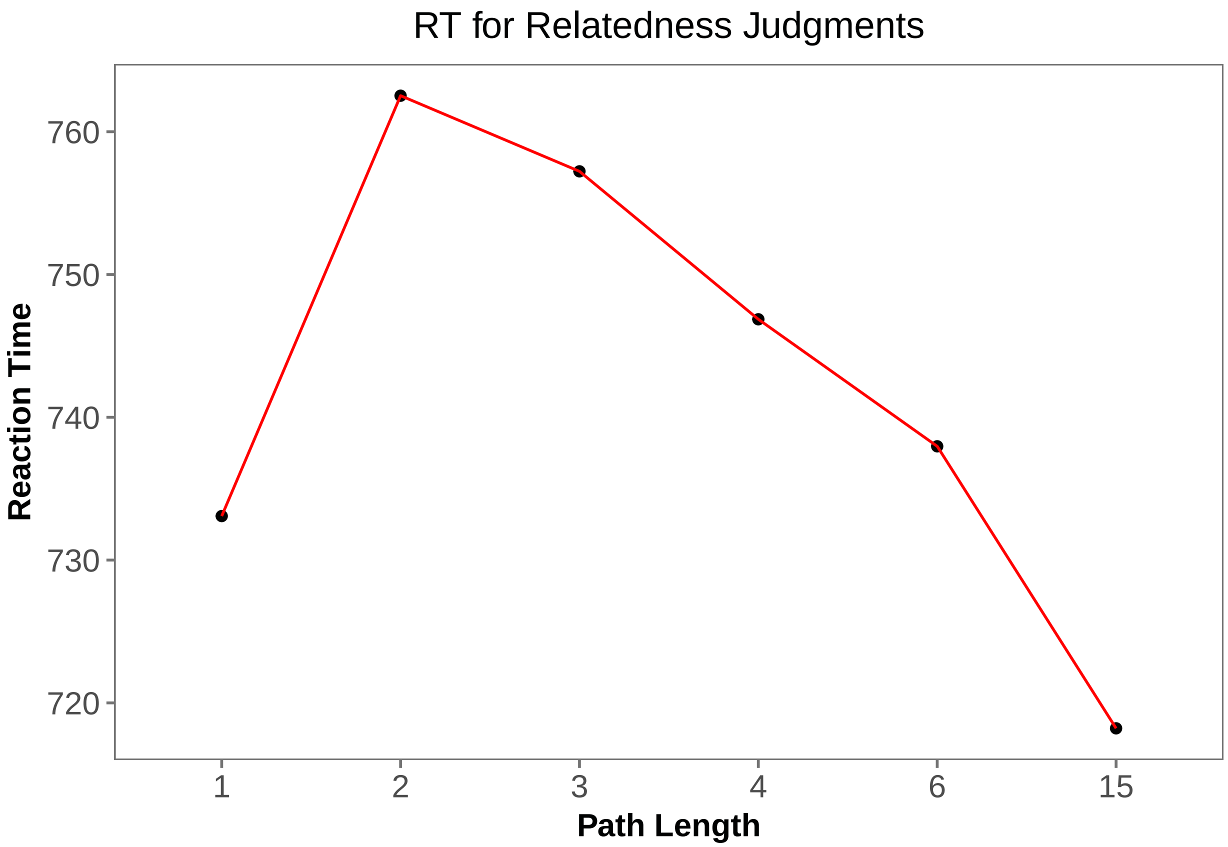
z-scored Reaction Time

First Trim

```
library(dplyr)
sem_firsttrim = sem %>% filter( rt > 250 & rt < 2000)
```

Raw RT aggregates After Trimming

```
sem_rt_firsttrim = group_by(sem_firsttrim, subject, pathlength ) %>%
  summarise_at(vars(rt), mean)

sem_rt_agg_firsttrim = group_by(sem_firsttrim, pathlength ) %>%
  summarise_at(vars(rt), mean)

sem_rt_agg_firsttrim$pathlengthfac = ordered(as.factor(as.character(sem_rt_agg_firsttrim$pathlength)),

library(ggplot2)
library(ggthemes)

sem_rt_agg_firsttrim %>%
  ggplot(aes(x = pathlengthfac, y = rt, group = 1))+
  geom_point()+
  geom_line(color = "red")+
   #geom_errorbar(aes(ymin=Trials - ci, ymax=Trials + ci),
  #           width=.2, color = "gray26",
   #          position = position_dodge(0.7))+
 theme_few()+
  #scale_x_continuous(breaks = c(1,2,3,4,6,15))+
    xlab("Path Length") + ylab("Reaction Time") +
  ggtitle("RT for Relatedness Judgments")  +
   theme(axis.text = element_text(size = rel(1)),
          axis.title = element_text(face = "bold", size = rel(1)),
          legend.title = element_text(face = "bold", size = rel(1)),
         plot.title = element_text(hjust = .5),
         strip.text.x = element_text(face = "bold", size = rel(1.4)))
```



Making the z-scores

```
## aggregate per subject all IVs and DVs
meanRT = group_by(sem_firsttrim, subject) %>%
  summarise_at(vars(rt), mean)
colnames(meanRT) = c("subject", "MeanRT")

sdRT = group_by(sem_firsttrim, subject) %>%
  summarise_at(vars(rt), sd)
colnames(sdRT) = c("subject", "sdRT")

RT_agg = merge(meanRT, sdRT, by = "subject")

## merge aggregate info with long data
sem_z = merge(sem_firsttrim, RT_agg, by = "subject", all.x = T)

## person and grand-mean centered scores using original and aggregate
library(dplyr)
sem_z = sem_z %>% mutate(zRT = (rt - MeanRT)/sdRT)

## checking: subject level means should be zero

sub_pic = group_by(sem_z, subject) %>%
  summarise_at(vars(zRT), mean)
```

Trimming z-RT

```r
sem_z_trimmed = subset(sem_z, sem_z$zRT < 3 & sem_z$zRT > -3)
```

Repeating z-scoring

```r
library(dplyr)

## aggregate per subject all IVs and DVs
meanRT_trim = group_by(sem_z_trimmed, subject) %>%
  summarise_at(vars(rt), mean)
colnames(meanRT_trim) = c("subject", "MeanRT_trim")

sdRT_trim = group_by(sem_z_trimmed, subject) %>%
  summarise_at(vars(rt), sd)
colnames(sdRT_trim) = c("subject", "sdRT_trim")

RT_agg_trim = merge(meanRT_trim, sdRT_trim, by = "subject")

## merge aggregate info with long data
new_sem_z = merge(sem_z_trimmed, RT_agg_trim, by = "subject", all.x = T)

## person and grand-mean centered scores using original and aggregate
library(dplyr)
new_sem_z = new_sem_z %>% mutate(zRT_trim = (rt - MeanRT_trim)/sdRT_trim)

## checking: subject level means should be zero

sub_pic = group_by(new_sem_z, subject) %>%
  summarise_at(vars(zRT_trim), mean)

#write.csv(new_pic_z, file="final_pic_z.csv")
```

Aggregating zRT

```r
z_sem_rt = group_by(new_sem_z, subject, pathlength ) %>%
  summarise_at(vars(zRT_trim, rt), mean)

z_sem_rt_rel = group_by(new_sem_z, subject, pathlength, Type ) %>%
  summarise_at(vars(zRT_trim, rt), mean)

z_sem_rt_agg = group_by(new_sem_z, pathlength ) %>%
  summarise_at(vars(zRT_trim), mean)

z_rmisc = Rmisc::summarySE(new_sem_z,
                        measurevar = "zRT_trim",
                        groupvars = c("pathlength"))

z_sem_rt_rel_agg = group_by(new_sem_z, pathlength, Type ) %>%
  summarise_at(vars(zRT_trim), mean)
```

```
z_sem_rt_rel_agg_rmisc = Rmisc::summarySE(new_sem_z,
                        measurevar = "zRT_trim",
                        groupvars = c("pathlength", "Type"))
```

ANOVA

```
z_sem_rt$pathlengthfac = ordered(as.factor(as.character(z_sem_rt$pathlength)),
                            levels = c("1", "2", "3", "4", "6", "15"))
z_sem_rt$subject = as.factor(z_sem_rt$subject)

z_rt_aov = aov(data = z_sem_rt, zRT_trim ~ pathlengthfac +
                     Error(subject/(pathlengthfac)))
summary(z_rt_aov)
```

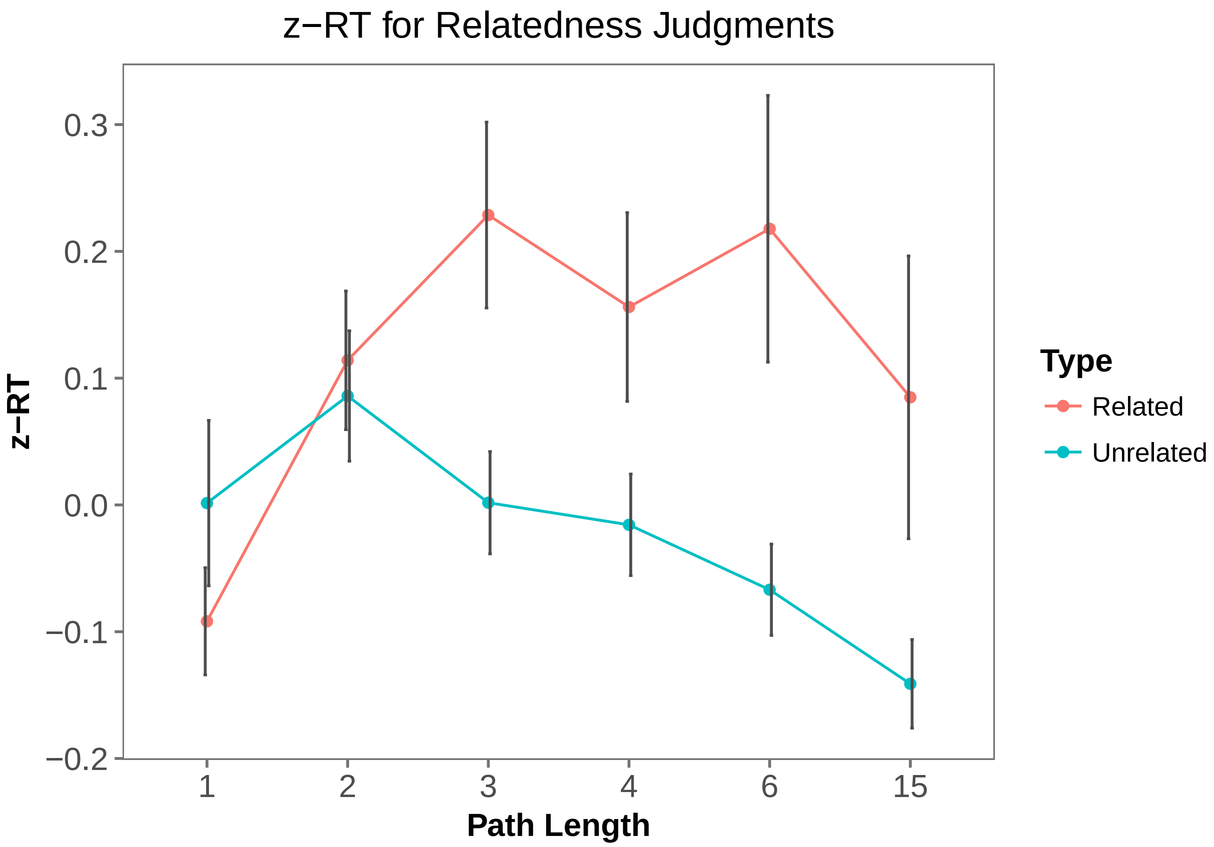
```
##
## Error: subject
##           Df  Sum Sq   Mean Sq F value Pr(>F)
## Residuals 79 0.00641 8.114e-05
##
## Error: subject:pathlengthfac
##                Df Sum Sq Mean Sq F value   Pr(>F)
## pathlengthfac   5  2.638  0.5276   13.45 3.99e-12 ***
## Residuals     395 15.494  0.0392
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
options(contrasts = c('contr.sum', 'contr.poly'))
 library(lsmeans)
 library(multcomp)
 sem_lsm = lsmeans::lsmeans(z_rt_aov, c("pathlengthfac"))
 prime_effect = cld(sem_lsm, alpha = 0.05,
                 adjust = "tukey", details = TRUE)
 library(knitr)
kable(subset(prime_effect$comparisons,prime_effect$comparisons$p.value < 0.1 ))
```

	contrast	estimate	SE	df	t.ratio	p.value
2	6 - 15	0.0894452	0.0313149	395	2.856311	0.0510422
4	4 - 15	0.1430209	0.0313149	395	4.567179	0.0000963
5	4 - 1	0.0850114	0.0313149	395	2.714725	0.0745850
7	3 - 15	0.1814771	0.0313149	395	5.795227	0.0000002
8	3 - 1	0.1234677	0.0313149	395	3.942774	0.0013288
9	3 - 6	0.0920319	0.0313149	395	2.938916	0.0404645
11	2 - 15	0.2181746	0.0313149	395	6.967114	0.0000000
12	2 - 1	0.1601652	0.0313149	395	5.114661	0.0000073
13	2 - 6	0.1287295	0.0313149	395	4.110803	0.0006793

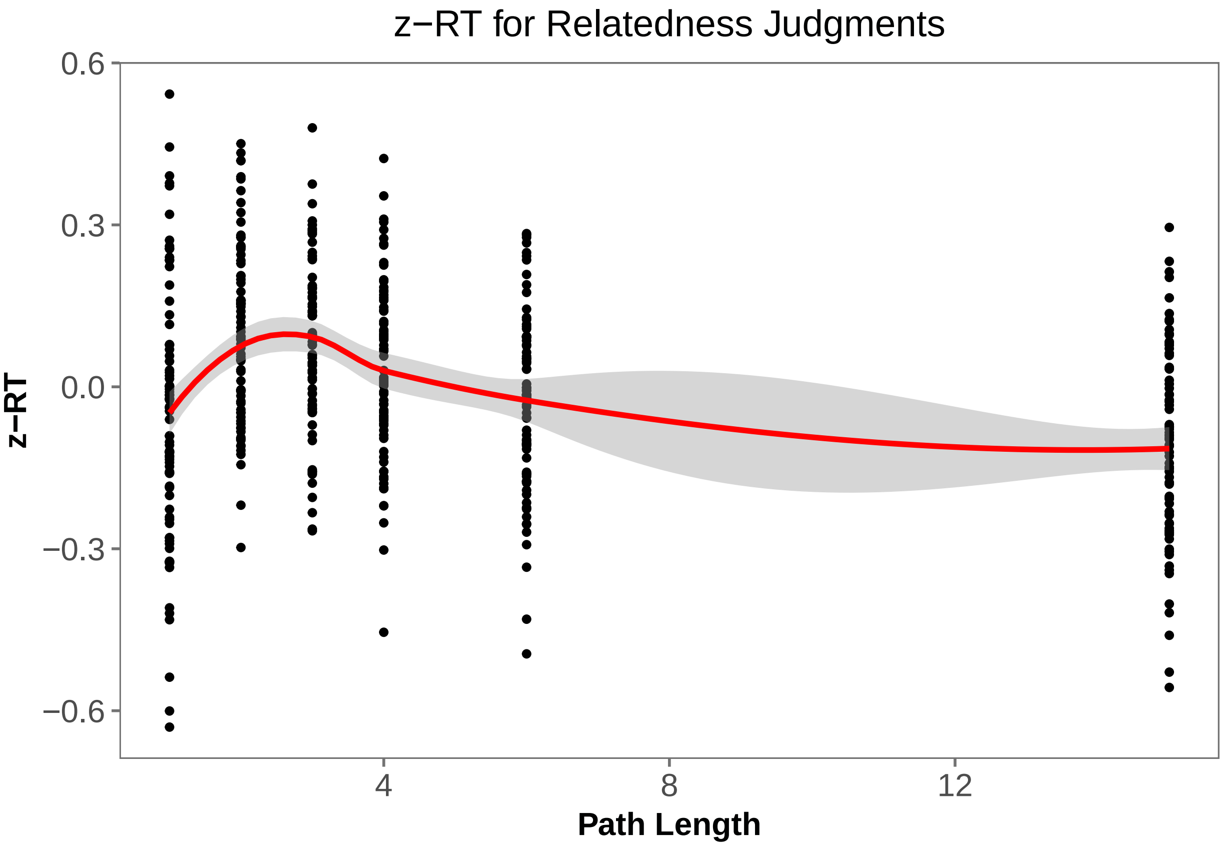
Plotting RTs: Rel-Unrel

```
z_sem_rt_rel_agg_rmisc$pathlengthfac = ordered(as.factor(as.character(z_sem_rt_rel_agg_rmisc$pathlength

z_sem_rt_rel_agg_rmisc$zRT_trim = as.numeric(z_sem_rt_rel_agg_rmisc$zRT_trim)

library(ggplot2)
library(ggthemes)

z_sem_rt_rel_agg_rmisc %>%
  ggplot(aes(x = pathlengthfac, y = zRT_trim, group = Type, color = Type))+
  geom_point()+
#  geom_smooth(method = "loess")+
geom_line()+
  geom_errorbar(aes(ymin=zRT_trim - ci, ymax=zRT_trim + ci),
             width=.05, color = "gray30",
             position = position_dodge(0.05))+
 theme_few()+
  #scale_x_continuous(breaks = c(1,2,3,4,5,6,10,15,20))+
    xlab("Path Length") + ylab("z-RT") +
  ggtitle("z-RT for Relatedness Judgments")  +
   theme(axis.text = element_text(size = rel(1)),
          axis.title = element_text(face = "bold", size = rel(1)),
          legend.title = element_text(face = "bold", size = rel(1)),
         plot.title = element_text(hjust = .5),
         strip.text.x = element_text(face = "bold", size = rel(1.4)))
```



Quadratic Trend

```
z_sem_rt %>%
  ggplot(aes(x = pathlength, y = zRT_trim))+
  geom_point(color = "black", size = 1)+
  geom_smooth(method = "loess", color = "red")+
   #geom_errorbar(aes(ymin=Trials - ci, ymax=Trials + ci),
  #           width=.2, color = "gray26",
   #          position = position_dodge(0.7))+
 theme_few()+
 # scale_x_continuous(breaks = c(1,2,3,4,5,6,10,15,20))+
    xlab("Path Length") + ylab("z-RT") +
  ggtitle("z-RT for Relatedness Judgments")  +
 # facet_wrap(~subject)+
   theme(axis.text = element_text(size = rel(1)),
          axis.title = element_text(face = "bold", size = rel(1)),
          legend.title = element_text(face = "bold", size = rel(1)),
         plot.title = element_text(hjust = .5),
         strip.text.x = element_text(face = "bold", size = rel(1.4)))
```



Regressions

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
sem$pathlengthfac = ordered(as.factor(as.character(sem$pathlength)),
                            levels = c("1", "2", "3", "4", "6","15"))

new_sem_z$pathlengthfac = ordered(as.factor(as.character(new_sem_z$pathlength)),
                            levels = c("1", "2", "3", "4", "6","15"))

m0 = lmer(data = new_sem_z, rt ~ 1 + (1|subject) + (1|trial_index) +
            (1|ItemNo))
summary(m0)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: rt ~ 1 + (1 | subject) + (1 | trial_index) + (1 | ItemNo)
##    Data: new_sem_z
##
## REML criterion at convergence: 249131.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.2009 -0.6212 -0.1612  0.4479  5.3272
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  ItemNo      (Intercept)  1522.3   39.02
##  trial_index (Intercept)   208.9   14.45
##  subject     (Intercept) 23823.6  154.35
##  Residual                50252.2  224.17
## Number of obs: 18170, groups:  ItemNo, 1200; trial_index, 240; subject, 80
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)    734.2       17.4   42.19
```

```
reghelper::ICC(m0)
```

```
## [1] 0.3371028
```

```
contrasts(sem$pathlengthfac) = contr.treatment(6, base = 1)


contrasts(new_sem_z$pathlengthfac) = contr.treatment(6, base = 2)

m1_fixed = lmer(data = new_sem_z, zRT_trim ~ pathlengthfac +
           (1|subject) + (1|ItemNo) + (1|trial_index))
summary(m1_fixed)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: zRT_trim ~ pathlengthfac + (1 | subject) + (1 | ItemNo) + (1 |
##     trial_index)
##    Data: new_sem_z
##
## REML criterion at convergence: 51296.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.5782 -0.7127 -0.1964  0.5389  3.9565
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  ItemNo      (Intercept) 0.033695 0.18356
##  trial_index (Intercept) 0.004246 0.06516
##  subject     (Intercept) 0.000000 0.00000
##  Residual                0.952788 0.97611
## Number of obs: 18170, groups:  ItemNo, 1200; trial_index, 240; subject, 80
##
## Fixed effects:
##                 Estimate Std. Error t value
## (Intercept)      0.09995    0.02245   4.451
## pathlengthfac1 -0.16082    0.03115  -5.162
## pathlengthfac3 -0.03054    0.03115  -0.980
## pathlengthfac4 -0.07032    0.03117  -2.256
## pathlengthfac5 -0.12080    0.03116  -3.876
## pathlengthfac6 -0.21266    0.03115  -6.827
##
## Correlation of Fixed Effects:
##             (Intr) pthln1 pthln3 pthln4 pthln5
## pthlngthfc1 -0.696
## pthlngthfc3 -0.696  0.502
## pthlngthfc4 -0.695  0.501  0.501
## pthlngthfc5 -0.695  0.501  0.501  0.501
## pthlngthfc6 -0.696  0.501  0.502  0.501  0.501
```

Concreteness Norms

```r
elpnorms = read.csv("ELP_norms.csv", header = TRUE, sep = ",")
elpnorms = elpnorms[,c(1,2)]
colnames(elpnorms) = c("prime_word", "prime_concreteness")
elpnorms$prime_word = toupper(elpnorms$prime_word)
elpnorms$prime_word = paste(elpnorms$prime_word, "")

elpnorms$prime_word = as.character(elpnorms$prime_word)

sem$prime_word = as.character(sem$prime_word)
merged_sem_prime= inner_join(sem, elpnorms, by = "prime_word")
merged_sem_prime = merged_sem_prime[,c(4,7,16,41)]

colnames(elpnorms) = c("target_word", "target_concreteness")
sem$target_word = as.character(sem$target_word)
merged_sem_target= inner_join(sem, elpnorms, by = "target_word")
merged_sem_target = merged_sem_target[,c(4,7, 17,41)]

merged_concretness = full_join(merged_sem_prime, merged_sem_target,
                           by = c("trial_index", "subject"))
merged_concretness$mean_conc = (merged_concretness$prime_concreteness +
                                merged_concretness$target_concreteness) / 2

### NOW WE HAVE CONCRETENESS NORMS FOR ALL ITEMS IN THE DATASET
### NEED TO COMBINE THIS WITH ACTUAL SEM DATA

new_sem_z$prime_word = as.character(new_sem_z$prime_word)
new_sem_z$target_word = as.character(new_sem_z$target_word)
final_sem = inner_join(new_sem_z, merged_concretness,
                      by = c("trial_index", "subject",
                             "prime_word", "target_word") )
```

Conreteness Model

```r
final_sem$pathlengthfac = ordered(as.factor(as.character(final_sem$pathlength)),
                           levels = c("1", "2", "3", "4", "6","15"))

elp_model = lmer(data = final_sem, zRT_trim ~ mean_len + mean_logf +
                   mean_ldtz + mean_conc +
           (1|subject) + (1|trial_index) + (1|ItemNo))
fit_from_elp = broom::augment(elp_model,final_sem)

contrasts(fit_from_elp$pathlengthfac) = contr.treatment(6, base = 2)

m1_fixed_elp = lmer(data = fit_from_elp, .resid ~ pathlengthfac +
           (1|subject) + (1|trial_index) + (1|ItemNo))
 summary(m1_fixed_elp)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: .resid ~ pathlengthfac + (1 | subject) + (1 | trial_index) +
##     (1 | ItemNo)
##    Data: fit_from_elp
##
## REML criterion at convergence: 45623.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.5871 -0.7167 -0.1965  0.5433  4.0644
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  ItemNo      (Intercept) 0.0000   0.0000
##  trial_index (Intercept) 0.0000   0.0000
##  subject     (Intercept) 0.0000   0.0000
##  Residual                0.9223   0.9604
## Number of obs: 16537, groups:  ItemNo, 1083; trial_index, 240; subject, 80
##
## Fixed effects:
##                 Estimate Std. Error t value
## (Intercept)      0.07081    0.01852   3.824
## pathlengthfac1 -0.11025    0.02595  -4.248
## pathlengthfac3 -0.02842    0.02612  -1.088
## pathlengthfac4 -0.05489    0.02584  -2.124
## pathlengthfac5 -0.08235    0.02622  -3.141
## pathlengthfac6 -0.14519    0.02588  -5.611
##
```

```
## Correlation of Fixed Effects:
##             (Intr) pthln1 pthln3 pthln4 pthln5
## pthlngthfc1 -0.714
## pthlngthfc3 -0.709  0.506
## pthlngthfc4 -0.717  0.511  0.508
## pthlngthfc5 -0.706  0.504  0.501  0.506
## pthlngthfc6 -0.716  0.511  0.507  0.513  0.505
```

```
contrasts(final_sem$pathlengthfac) = contr.treatment(6, base = 2)

m1_all_elp = lme4::lmer(data = final_sem, zRT_trim ~ pathlengthfac +
                    mean_len + mean_logf + mean_ldtz + mean_conc +
          (1|subject) + (1|trial_index) + (1|ItemNo))
summary(m1_all_elp)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: zRT_trim ~ pathlengthfac + mean_len + mean_logf + mean_ldtz +
##     mean_conc + (1 | subject) + (1 | trial_index) + (1 | ItemNo)
##    Data: final_sem
##
## REML criterion at convergence: 46604.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.5472 -0.7097 -0.1954  0.5384  4.0176
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  ItemNo      (Intercept) 0.029087 0.17055
##  trial_index (Intercept) 0.004143 0.06436
##  subject     (Intercept) 0.000000 0.00000
##  Residual                0.949705 0.97453
## Number of obs: 16537, groups:  ItemNo, 1083; trial_index, 240; subject, 80
##
## Fixed effects:
##                 Estimate Std. Error t value
## (Intercept)     0.338792   0.110868   3.056
## pathlengthfac1 -0.172789   0.031999  -5.400
## pathlengthfac3 -0.044592   0.032170  -1.386
## pathlengthfac4 -0.085183   0.031884  -2.672
## pathlengthfac5 -0.128964   0.032330  -3.989
## pathlengthfac6 -0.227538   0.031932  -7.126
## mean_len        0.015092   0.007331   2.059
## mean_logf      -0.008914   0.008808  -1.012
## mean_ldtz       0.006851   0.068955   0.099
## mean_conc      -0.065952   0.011881  -5.551
##
## Correlation of Fixed Effects:
##             (Intr) pthln1 pthln3 pthln4 pthln5 pthln6 men_ln mn_lgf mn_ldt
## pthlngthfc1 -0.129
## pthlngthfc3 -0.175  0.505
## pthlngthfc4 -0.186  0.508  0.507
## pthlngthfc5 -0.175  0.503  0.501  0.505
## pthlngthfc6 -0.165  0.510  0.507  0.511  0.506
## mean_len    -0.590 -0.013  0.027  0.022  0.040  0.032
## mean_logf   -0.657  0.015  0.024  0.037  0.032  0.026 -0.012
## mean_ldtz    0.006  0.020 -0.017  0.012  0.009  0.003 -0.360  0.550
## mean_conc   -0.666 -0.044 -0.007  0.022 -0.015 -0.029  0.240  0.311  0.111
```

```
 # m1_all_elp_random = lme4::lmer(data = final_sem, zRT_trim ~ pathlengthfac +
 #                      mean_len + mean_logf + mean_ldtz + mean_conc +
 #            (pathlengthfac|subject) +
 #               + (1|trial_index) + (pathlengthfac|ItemNo),
 #    control=lmerControl(optimizer="bobyqa",
 #            optCtrl=list(maxfun=1000000)))
 # summary(m1_all_elp_random)
 #
 ## centering so that contrasts are easier
final_sem$mean_len_c = scale(final_sem$mean_len,
                             center = TRUE, scale = FALSE)
final_sem$mean_logf_c = scale(final_sem$mean_logf,
                              center = TRUE, scale = FALSE)
final_sem$mean_ldtz_c = scale(final_sem$mean_ldtz,
                              center = TRUE, scale = FALSE)
final_sem$mean_conc_c = scale(final_sem$mean_conc,
                              center = TRUE, scale = FALSE)

 m1_all_elp_type = lme4::lmer(data = final_sem,
                              zRT_trim ~ pathlengthfac*Type +
                     mean_len_c + mean_logf_c + mean_ldtz_c + mean_conc_c +
           (1|subject) + (1|trial_index) +
              + (1|target_word))
 summary(m1_all_elp_type)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: zRT_trim ~ pathlengthfac * Type + mean_len_c + mean_logf_c +
##     mean_ldtz_c + mean_conc_c + (1 | subject) + (1 | trial_index) +
##     +(1 | target_word)
##    Data: final_sem
##
## REML criterion at convergence: 46602.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.5531 -0.7095 -0.1900  0.5451  4.1342
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  target_word (Intercept) 0.022649 0.15050
##  trial_index (Intercept) 0.003983 0.06311
##  subject     (Intercept) 0.000000 0.00000
##  Residual                0.951695 0.97555
## Number of obs: 16537, groups:
## target_word, 1741; trial_index, 240; subject, 80
##
## Fixed effects:
##                       Estimate Std. Error t value
## (Intercept)           0.106739   0.020964   5.092
## pathlengthfac1       -0.155567   0.029649  -5.247
## pathlengthfac3       -0.001377   0.030137  -0.046
## pathlengthfac4       -0.055803   0.030267  -1.844
## pathlengthfac5       -0.036590   0.034248  -1.068
## pathlengthfac6       -0.158174   0.035912  -4.404
## Type1                 0.007252   0.019361   0.375
## mean_len_c            0.013107   0.006762   1.938
## mean_logf_c          -0.009348   0.008132  -1.150
## mean_ldtz_c           0.019853   0.063358   0.313
## mean_conc_c          -0.065925   0.010924  -6.035
## pathlengthfac1:Type1 -0.054857   0.028067  -1.954
## pathlengthfac3:Type1  0.096917   0.028600   3.389
## pathlengthfac4:Type1  0.059490   0.028697   2.073
## pathlengthfac5:Type1  0.122251   0.032959   3.709
## pathlengthfac6:Type1  0.087595   0.034554   2.535
```

```
##
## Correlation matrix not shown by default, as p = 16 > 12.
## Use print(x, correlation=TRUE)  or
##   vcov(x)     if you need it
```

```
m1_all_elp_type_withoutcov = lme4::lmer(data = final_sem,
                             zRT_trim ~ pathlengthfac*Type +
          (1|subject) + (1|trial_index) +
             + (1|target_word))
summary(m1_all_elp_type_withoutcov)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## zRT_trim ~ pathlengthfac * Type + (1 | subject) + (1 | trial_index) +
##     +(1 | target_word)
##    Data: final_sem
##
## REML criterion at convergence: 49782.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.6843 -0.7105 -0.1878  0.5432  4.0863
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  target_word (Intercept) 0.026296 0.16216
##  trial_index (Intercept) 0.004901 0.07001
##  subject     (Intercept) 0.000000 0.00000
##  Residual                0.954268 0.97687
## Number of obs: 17636, groups:
## target_word, 1845; trial_index, 240; subject, 80
##
## Fixed effects:
##                       Estimate Std. Error t value
## (Intercept)           0.097171   0.020739   4.686
## pathlengthfac1       -0.150061   0.029088  -5.159
## pathlengthfac3        0.008553   0.029403   0.291
## pathlengthfac4       -0.033909   0.029833  -1.137
## pathlengthfac5       -0.023459   0.033135  -0.708
## pathlengthfac6       -0.137580   0.035110  -3.919
## Type1                 0.018972   0.018866   1.006
## pathlengthfac1:Type1 -0.057105   0.027338  -2.089
## pathlengthfac3:Type1  0.086727   0.027707   3.130
## pathlengthfac4:Type1  0.053802   0.028094   1.915
## pathlengthfac5:Type1  0.119827   0.031672   3.783
## pathlengthfac6:Type1  0.079640   0.033608   2.370
##
## Correlation of Fixed Effects:
##             (Intr) pthln1 pthln3 pthln4 pthln5 pthln6 Type1  pt1:T1 pt3:T1
## pthlngthfc1 -0.673
## pthlngthfc3 -0.664  0.475
## pthlngthfc4 -0.655  0.470  0.465
## pthlngthfc5 -0.587  0.419  0.414  0.409
## pthlngthfc6 -0.557  0.398  0.393  0.390  0.347
## Type1        0.033 -0.024 -0.023 -0.023 -0.020 -0.020
## pthlngt1:T1 -0.022 -0.158  0.014  0.015  0.014  0.013 -0.690
## pthlngt3:T1 -0.022  0.015  0.216  0.015  0.012  0.014 -0.680  0.470
## pthlngt4:T1 -0.023  0.016  0.016  0.261  0.013  0.013 -0.671  0.464  0.457
## pthlngt5:T1 -0.019  0.015  0.013  0.013  0.433  0.012 -0.595  0.411  0.405
## pthlngt6:T1 -0.019  0.013  0.013  0.014  0.010  0.507 -0.561  0.387  0.382
##             pt4:T1 pt5:T1
## pthlngthfc1
## pthlngthfc3
## pthlngthfc4
## pthlngthfc5
## pthlngthfc6
## Type1
## pthlngt1:T1
## pthlngt3:T1
## pthlngt4:T1
## pthlngt5:T1  0.399
## pthlngt6:T1  0.377  0.333
```

Item Diagnostics

```
item_mean = group_by(new_sem_z, pathlength, ItemNo) %>%
  summarise_at(vars(zRT_trim), mean)

item_rel = group_by(sem, pathlength, ItemNo, Type) %>%
  summarise(Trials = n())

item_rel_rt = group_by(sem, pathlength, ItemNo, Type) %>%
  summarise_at(vars(rt), mean)

item_rt_collapsed = group_by(sem, pathlength, ItemNo) %>%
  summarise_at(vars(rt), mean)

item_rel_wide = tidyr::spread(item_rel, Type, Trials)

item_rel_rt_wide = tidyr::spread(item_rel_rt, Type, rt)

item_sub = group_by(sem, ItemNo, subject) %>%
  summarise(Subjects = n())
```

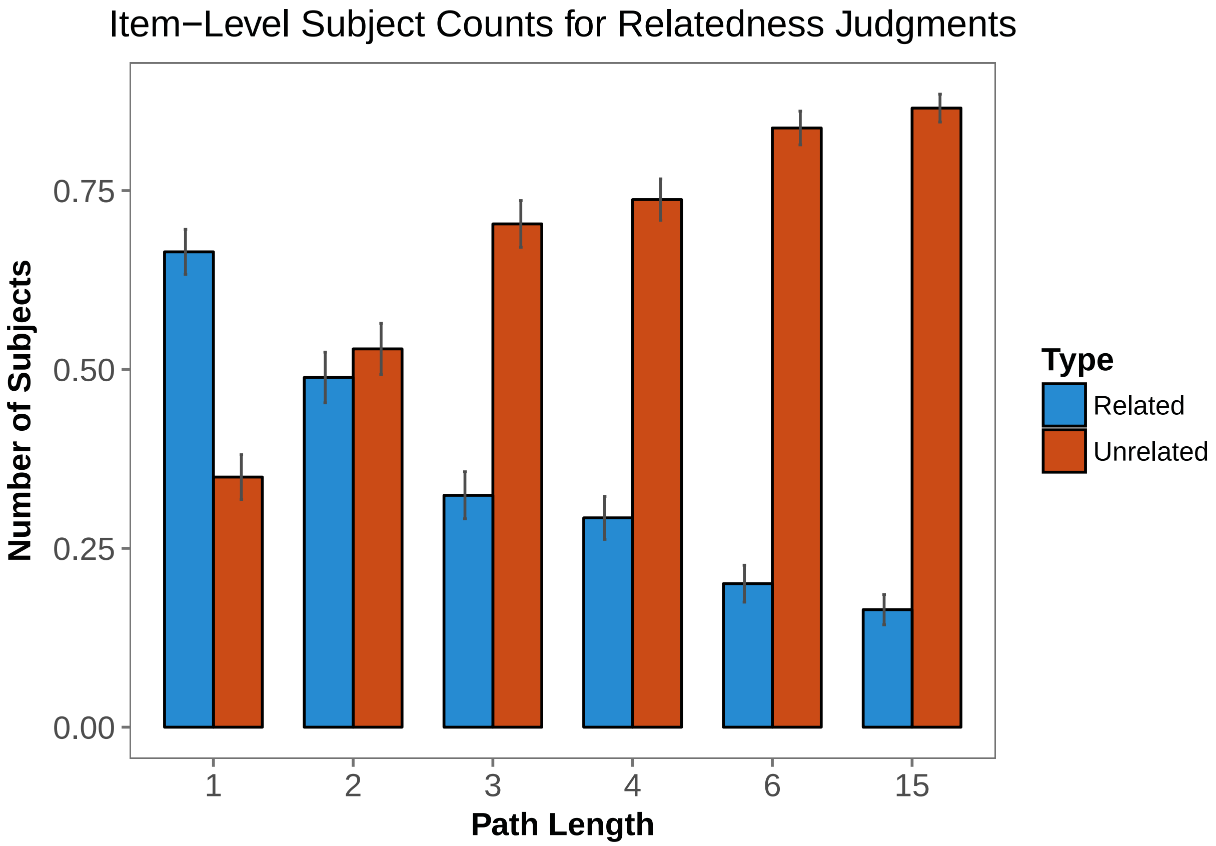
```
item_rel_wide$Related  = ifelse(is.na(item_rel_wide$Related), 0, item_rel_wide$Related)

item_rel_wide$Unrelated  = ifelse(is.na(item_rel_wide$Unrelated), 0, item_rel_wide$Unrelated)

item_rel_rt_wide$Related  = ifelse(is.na(item_rel_rt_wide$Related), 0, item_rel_rt_wide$Related)

item_rel_rt_wide$Unrelated  = ifelse(is.na(item_rel_rt_wide$Unrelated), 0, item_rel_rt_wide$Unrelated)
```

Item Level Analyses

Proportion of Subjects

Plot

```
## we have item_rel_wide and item_rel_rt_wide with RTs and number of subjects
## who called these items related/unrelated for each path length

item_rel_main = merge(item_rel, item_rel_rt,
                      by = c("pathlength", "ItemNo", "Type"))
colnames(item_rel_main) =c ("pathlength", "ItemNo",
                            "Type", "Trials","rt")

itemrt_agg = Rmisc::summarySE(item_rel_main,
                        measurevar = "rt",
                        groupvars = c("pathlength", "Type"))

itemrt_agg_collapsed = Rmisc::summarySE(item_rt_collapsed,
                        measurevar = "rt",
                        groupvars = c("pathlength"))

item_rel_main$Proportion = item_rel_main$Trials/16


item_agg = Rmisc::summarySE(item_rel_main,
                        measurevar = "Proportion",
                        groupvars = c("pathlength", "Type"))
item_agg$pathlengthfac = ordered(as.factor(as.character(item_agg$pathlength)),
                            levels = c("1", "2", "3", "4", "6", "15"))

item_agg %>%
  ggplot(aes(x = pathlengthfac, y = Proportion, group = Type, fill = Type))+
#  geom_smooth(method = "loess")+
 geom_bar(stat = "identity", position = "dodge", width = 0.7, color= "black")+
  geom_errorbar(aes(ymin=Proportion - ci, ymax=Proportion + ci),
             width=.05, color = "gray30",
             position = position_dodge(0.8))+
 theme_few()+
  scale_fill_solarized()+
  #scale_x_continuous(breaks = c(1,2,3,4,5,6,10,15,20))+
    xlab("Path Length") + ylab("Number of Subjects") +
  ggtitle("Item-Level Subject Counts for Relatedness Judgments")  +
   theme(axis.text = element_text(size = rel(1)),
          axis.title = element_text(face = "bold", size = rel(1)),
          legend.title = element_text(face = "bold", size = rel(1)),
         plot.title = element_text(hjust = .5),
         strip.text.x = element_text(face = "bold", size = rel(1.4)))
```



ANOVA

```
## item_rel_wide has 1200 rows with 0s for items. convert that into wide.

library(tidyr)
```

```
##
## Attaching package: 'tidyr'

## The following object is masked from 'package:Matrix':
##
##     expand
```

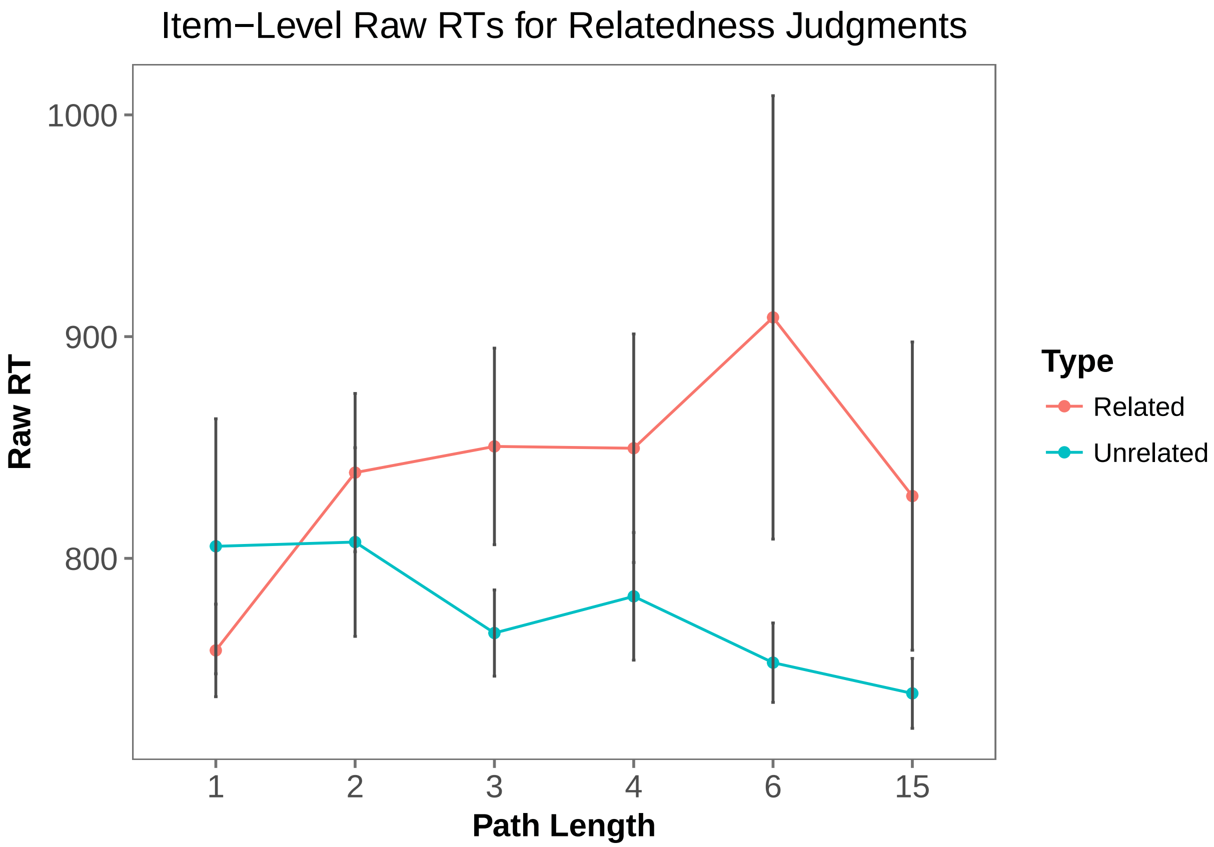
```
item_prop_data = item_rel_wide %>%
  gather(Type, Trials, Related:Unrelated)

item_prop_data$Proportion = item_prop_data$Trials/16

item_prop_data = item_prop_data[order(item_prop_data$ItemNo),]
```

```
item_prop_data$pathlengthfac = ordered(as.factor(as.character(item_prop_data$pathlength)),
                            levels = c("1", "2", "3", "4", "6", "15"))
library(lme4)
itemrelunrel_aov = aov(data = item_prop_data,
                       Proportion ~ pathlengthfac*Type +
                     Error(ItemNo/(Type)))
summary(itemrelunrel_aov)
```

```
##
## Error: ItemNo
##                  Df    Sum Sq   Mean Sq F value Pr(>F)
## pathlengthfac    5 2.900e-28 5.896e-29   0.317  0.903
## Residuals     1194 2.223e-25 1.862e-28
##
## Error: ItemNo:Type
##                      Df Sum Sq Mean Sq F value Pr(>F)
## Type                  1  66.54   66.54   750.4 <2e-16 ***
## pathlengthfac:Type    5  81.84   16.37   184.6 <2e-16 ***
## Residuals          1194 105.88    0.09
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Raw RT Plot Rel-Unrel

```
itemrt_agg$pathlengthfac = ordered(as.factor(as.character(itemrt_agg$pathlength)),
                            levels = c("1", "2", "3", "4", "6", "15"))

itemrt_agg %>%
  ggplot(aes(x = pathlengthfac, y = rt, group = Type, color = Type))+
#  geom_smooth(method = "loess")+
  geom_point()+
geom_line()+
  geom_errorbar(aes(ymin=rt - ci, ymax=rt + ci),
             width=.05, color = "gray30",
             position = position_dodge(0))+
 theme_few()+
  #scale_x_continuous(breaks = c(1,2,3,4,5,6,10,15,20))+
    xlab("Path Length") + ylab("Raw RT") +
  ggtitle("Item-Level Raw RTs for Relatedness Judgments")  +
   theme(axis.text = element_text(size = rel(1)),
          axis.title = element_text(face = "bold", size = rel(1)),
          legend.title = element_text(face = "bold", size = rel(1)),
         plot.title = element_text(hjust = .5),
         strip.text.x = element_text(face = "bold", size = rel(1.4)))
```



RT ANOVA Rel-Unrel

```
item_rel_main$pathlengthfac = ordered(as.factor(as.character(item_rel_main$pathlength)),
                              levels = c("1", "2", "3", "4", "6", "15"))
contrasts(item_rel_main$pathlengthfac) = contr.treatment(6, base = 2)
contrasts(item_rel_main$Type) = contr.treatment(2, base = 2)


item_rt_aov = lme4::lmer(data = item_rel_main, rt ~ pathlengthfac*Type +
                            (1|ItemNo))
summary(item_rt_aov)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: rt ~ pathlengthfac * Type + (1 | ItemNo)
##    Data: item_rel_main
##
## REML criterion at convergence: 32510.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.5813 -0.4281 -0.1238  0.2605 22.1734
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  ItemNo   (Intercept)  1094     33.07
##  Residual             97675    312.53  
## Number of obs: 2275, groups:  ItemNo, 1200
## 
## Fixed effects:
##                       Estimate Std. Error t value
## (Intercept)            807.339     22.335   36.15
## pathlengthfac1          -1.913     31.832   -0.06
## pathlengthfac3         -41.021     31.507   -1.30
## pathlengthfac4         -24.484     31.546   -0.78
## pathlengthfac5         -54.385     31.507   -1.73
## pathlengthfac6         -68.240     31.507   -2.17
## Type1                   31.336     31.534    0.99
## pathlengthfac1:Type1   -78.278     44.629   -1.75
## pathlengthfac3:Type1    52.791     44.911    1.18
## pathlengthfac4:Type1    35.504     44.972    0.79
## pathlengthfac5:Type1   124.329     45.684    2.72
## pathlengthfac6:Type1    57.672     45.602    1.26
## 
## Correlation of Fixed Effects:
##             (Intr) pthln1 pthln3 pthln4 pthln5 pthln6 Type1  pt1:T1 pt3:T1
## pthlngthfc1 -0.702                                                        
## pthlngthfc3 -0.709  0.497                                                 
## pthlngthfc4 -0.708  0.497  0.502                                          
## pthlngthfc5 -0.709  0.497  0.503  0.502                                   
## pthlngthfc6 -0.709  0.497  0.503  0.502  0.503                            
## Type1       -0.701  0.492  0.497  0.496  0.497  0.497                     
## pthlngt1:T1  0.495 -0.706 -0.351 -0.350 -0.351 -0.351 -0.707              
## pthlngt3:T1  0.492 -0.345 -0.694 -0.348 -0.349 -0.349 -0.702  0.496       
## pthlngt4:T1  0.491 -0.345 -0.348 -0.694 -0.348 -0.348 -0.701  0.495  0.492
## pthlngt5:T1  0.484 -0.339 -0.343 -0.342 -0.682 -0.343 -0.690  0.488  0.485
## pthlngt6:T1  0.484 -0.340 -0.343 -0.343 -0.343 -0.683 -0.692  0.489  0.486
##             pt4:T1 pt5:T1
## pthlngthfc1              
## pthlngthfc3              
## pthlngthfc4              
## pthlngthfc5              
## pthlngthfc6              
## Type1                    
## pthlngt1:T1              
## pthlngt3:T1              
## pthlngt4:T1              
## pthlngt5:T1  0.484       
## pthlngt6:T1  0.485  0.477
```

```
car::Anova(item_rt_aov)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
## 
## Response: rt
##                      Chisq Df Pr(>Chisq)    
## pathlengthfac       7.5307  5  0.1840642    
## Type               21.8948  1   2.88e-06 ***
## pathlengthfac:Type 22.0476  5  0.0005128 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Word2vec Model

```
word2vec = read.csv("word2veccosines.csv", header = TRUE, sep = ",")

final_word2vec = merge(final_sem, word2vec,
                       by = c( "ItemNo", "proc"))
final_word2vec$mean_conc_c = as.numeric(final_word2vec$mean_conc_c)
final_word2vec$mean_len_c = as.numeric(final_word2vec$mean_len_c)
final_word2vec$mean_logf_c = as.numeric(final_word2vec$mean_logf_c)
final_word2vec$mean_ldtz_c = as.numeric(final_word2vec$mean_ldtz_c)

final_word2vec = final_word2vec %>% arrange(subject, ItemNo)

m1_word2vec = lme4::lmer(data = final_word2vec,
                              zRT_trim ~ word2veccosine*Type +
                     mean_len + mean_logf + mean_ldtz + mean_conc +
           (1|subject) + (1|trial_index) +
              + (1|target_word))
 summary(m1_word2vec)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## zRT_trim ~ word2veccosine * Type + mean_len + mean_logf + mean_ldtz +
##     mean_conc + (1 | subject) + (1 | trial_index) + +(1 | target_word)
##    Data: final_word2vec
##
## REML criterion at convergence: 46517.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.6065 -0.7115 -0.1889  0.5376  4.1045
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  target_word (Intercept) 0.022689 0.15063
##  trial_index (Intercept) 0.003642 0.06035
##  subject     (Intercept) 0.000000 0.00000
##  Residual                0.951902 0.97565
## Number of obs: 16521, groups:
## target_word, 1740; trial_index, 240; subject, 80
##
## Fixed effects:
##                       Estimate Std. Error t value
## (Intercept)           0.353044   0.101052   3.494
## word2veccosine       -0.220758   0.063011  -3.503
## Type1                 0.165491   0.013631  12.141
## mean_len              0.013886   0.006758   2.055
## mean_logf            -0.009849   0.008141  -1.210
## mean_ldtz             0.030085   0.063340   0.475
## mean_conc            -0.067923   0.010907  -6.227
## word2veccosine:Type1 -0.604637   0.060164 -10.050
##
## Correlation of Fixed Effects:
```

```
##             (Intr) wrd2vc Type1  men_ln mn_lgf mn_ldt mn_cnc
## word2veccsn -0.131
## Type1        0.095 -0.190
## mean_len    -0.596  0.010 -0.038
## mean_logf   -0.669  0.068 -0.026 -0.011
## mean_ldtz    0.010  0.008  0.021 -0.365  0.546
## mean_conc   -0.667 -0.056  0.027  0.237  0.309  0.112
## wrd2vccs:T1 -0.016 -0.111 -0.740 -0.013 -0.010  0.000  0.005
```

```
m1_pathlength = lme4::lmer(data = final_word2vec,
                             zRT_trim ~ pathlengthfac*Type +
                    mean_len + mean_logf + mean_ldtz + mean_conc +
          (1|subject) + (1|trial_index) +
             + (1|target_word))
summary(m1_pathlength)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## zRT_trim ~ pathlengthfac * Type + mean_len + mean_logf + mean_ldtz +
##     mean_conc + (1 | subject) + (1 | trial_index) + +(1 | target_word)
##    Data: final_word2vec
##
## REML criterion at convergence: 46558.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.5535 -0.7096 -0.1901  0.5443  4.1324
##
## Random effects:
##  Groups      Name        Variance Std.Dev.
##  target_word (Intercept) 0.022784 0.15094
##  trial_index (Intercept) 0.003895 0.06241
##  subject     (Intercept) 0.000000 0.00000
##  Residual                0.951712 0.97556
## Number of obs: 16521, groups:
## target_word, 1740; trial_index, 240; subject, 80
##
## Fixed effects:
##                         Estimate Std. Error t value
## (Intercept)             0.359990   0.102484   3.513
## pathlengthfac1         -0.155496   0.029706  -5.234
## pathlengthfac3         -0.001274   0.030188  -0.042
## pathlengthfac4         -0.055744   0.030317  -1.839
## pathlengthfac5         -0.036486   0.034299  -1.064
## pathlengthfac6         -0.158137   0.035958  -4.398
## Type1                   0.007326   0.019416   0.377
## mean_len                0.013089   0.006767   1.934
## mean_logf              -0.009357   0.008136  -1.150
## mean_ldtz               0.019838   0.063407   0.313
## mean_conc              -0.065939   0.010937  -6.029
## pathlengthfac1:Type1 -0.054875   0.028107  -1.952
## pathlengthfac3:Type1  0.096857   0.028639   3.382
## pathlengthfac4:Type1  0.059397   0.028736   2.067
## pathlengthfac5:Type1  0.122147   0.032996   3.702
## pathlengthfac6:Type1  0.087521   0.034586   2.531
```

```
##
## Correlation matrix not shown by default, as p = 16 > 12.
## Use print(x, correlation=TRUE)  or
##   vcov(x)     if you need it
```

```
anova(m1_pathlength,m1_word2vec)
```

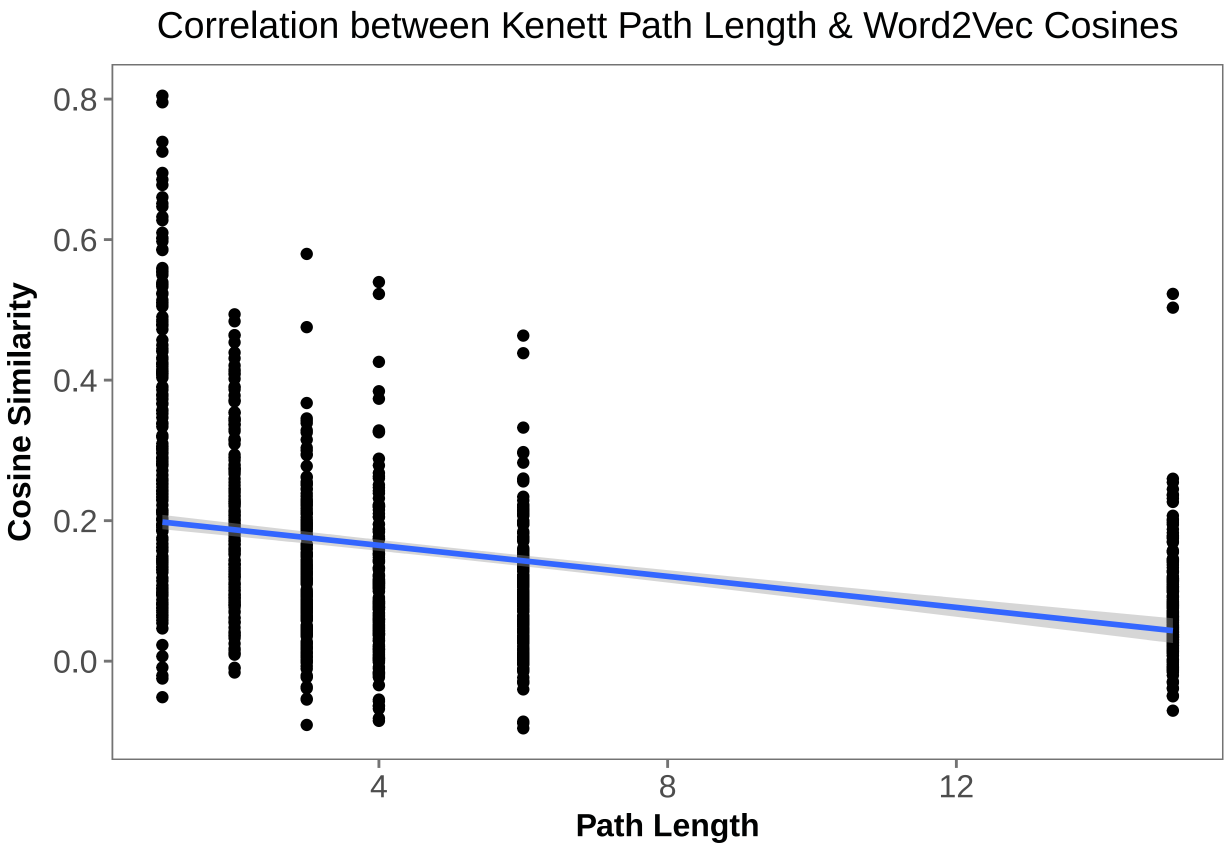
```
## refitting model(s) with ML (instead of REML)
```

```
## Data: final_word2vec
## Models:
## m1_word2vec: zRT_trim ~ word2veccosine * Type + mean_len + mean_logf + mean_ldtz +
## m1_word2vec:     mean_conc + (1 | subject) + (1 | trial_index) + +(1 | target_word)
## m1_pathlength: zRT_trim ~ pathlengthfac * Type + mean_len + mean_logf + mean_ldtz +
## m1_pathlength:     mean_conc + (1 | subject) + (1 | trial_index) + +(1 | target_word)
##               Df   AIC   BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
## m1_word2vec   12 46492 46584 -23234    46468
## m1_pathlength 20 46501 46655 -23230    46461 6.5237      8     0.5888
```

ScatterPlot

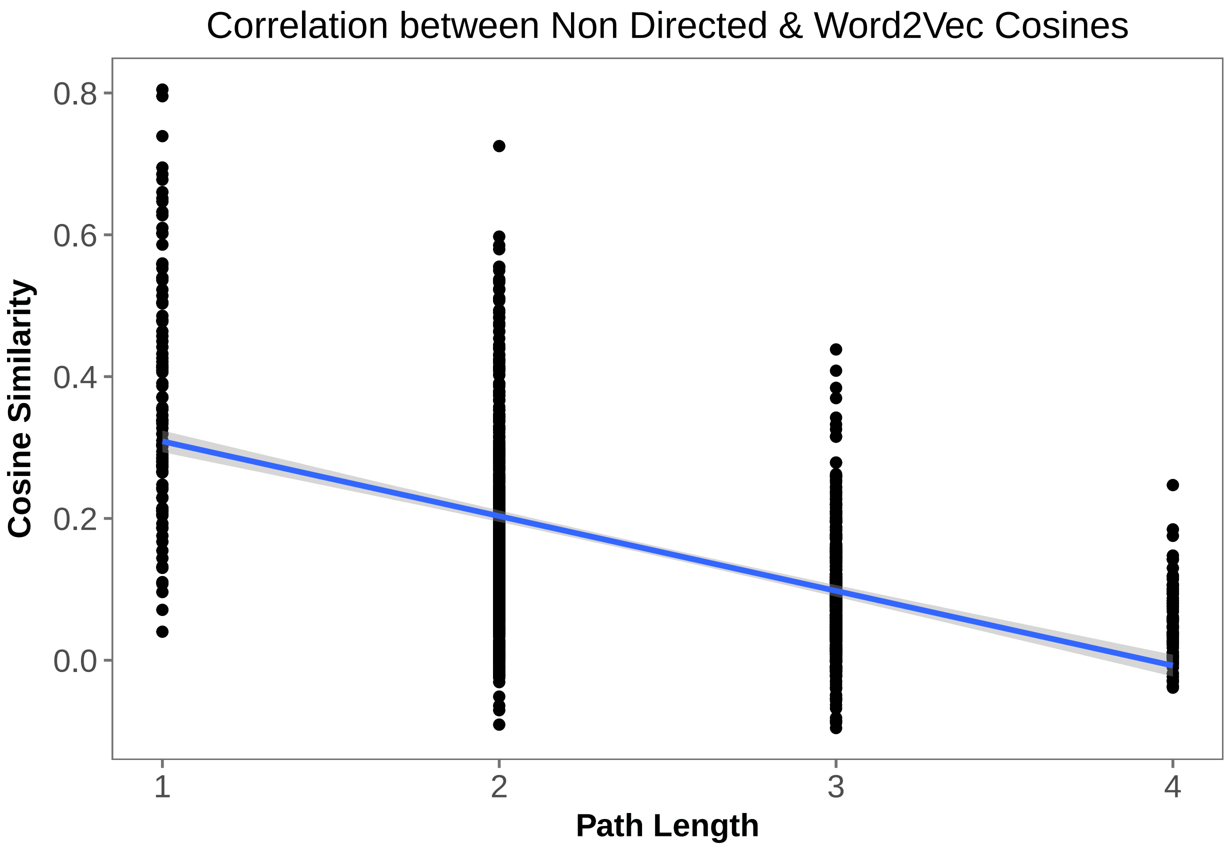
Kenett and Word2Vec

```
item_word2vec = group_by(final_word2vec, ItemNo) %>%
  summarise_at(vars(pathlength,undirected, word2veccosine), mean)
item_word2vec$pathlengthfac = ordered(as.factor(as.character(item_word2vec$pathlength)),
                           levels = c("1", "2", "3", "4", "6", "15"))
ggplot(item_word2vec,
       aes(x = pathlength, y = word2veccosine))+
geom_point()+
  geom_smooth(method = "lm")+
 theme_few()+
    xlab("Path Length") + ylab("Cosine Similarity") +
  ggtitle("Correlation between Kenett Path Length & Word2Vec Cosines")  +
   theme(axis.text = element_text(size = rel(1)),
          axis.title = element_text(face = "bold", size = rel(1)),
          legend.title = element_text(face = "bold", size = rel(1)),
         plot.title = element_text(hjust = .5),
         strip.text.x = element_text(face = "bold", size = rel(1.4)))
```



Kenett and Word2Vec

```
ggplot(item_word2vec,
       aes(x = undirected, y = word2veccosine))+
geom_point()+
  geom_smooth(method = "lm")+
 theme_few()+
    xlab("Path Length") + ylab("Cosine Similarity") +
  ggtitle("Correlation between Non Directed & Word2Vec Cosines")  +
   theme(axis.text = element_text(size = rel(1)),
          axis.title = element_text(face = "bold", size = rel(1)),
          legend.title = element_text(face = "bold", size = rel(1)),
         plot.title = element_text(hjust = .5),
         strip.text.x = element_text(face = "bold", size = rel(1.4)))
```


Correlation between Non Directed & Word2Vec Cosines
Cosine Similarity
0.8
0.6
0.4
0.2
0.0
Path Length
1
2
3
4