Network Demasking

Reading the Data

```
library(dplyr)
netdemask = read.csv("Compiled_NetworksE3.csv", header = TRUE, sep = ",")
## filtering out incorrect target responses
netdemask = netdemask %>% filter(TargetAccuracy == 1)
```

Raw Reaction Time

```
netdemask_rt = group_by(netdemask, subject, pathlength ) %>%
  summarise_at(vars(RTRecogniseTarget), mean)

netdemask_rt_agg = Rmisc::summarySE(netdemask_rt,
                        measurevar = "RTRecogniseTarget",
                        groupvars = c("pathlength"))
```

ANOVA

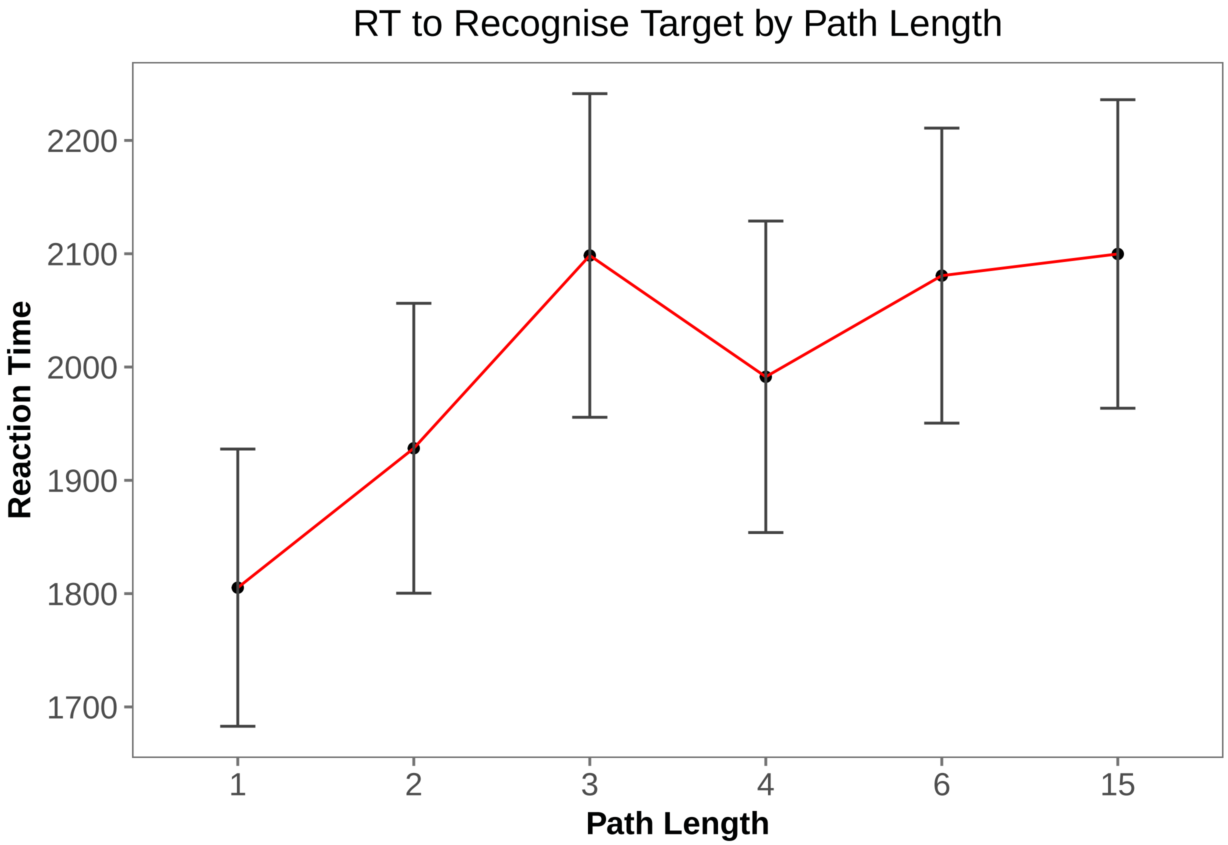
```
netdemask_rt$pathlengthfac = ordered(as.factor(as.character(netdemask_rt$pathlength)),
                             levels = c("1", "2", "3", "4", "6", "15"))
netdemask_rt$subject = as.factor(netdemask_rt$subject)
rt_aov = aov(data = netdemask_rt, RTRecogniseTarget ~ pathlengthfac +
            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

```
##
## Error: subject
##           Df  Sum Sq Mean Sq F value Pr(>F)
## Residuals  9 9039609 1004401
##
## Error: subject:pathlengthfac
##               Df Sum Sq Mean Sq F value   Pr(>F)
## pathlengthfac  5 693001  138600   12.18 1.72e-07 ***
## Residuals     45 511956   11377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Plotting RTs

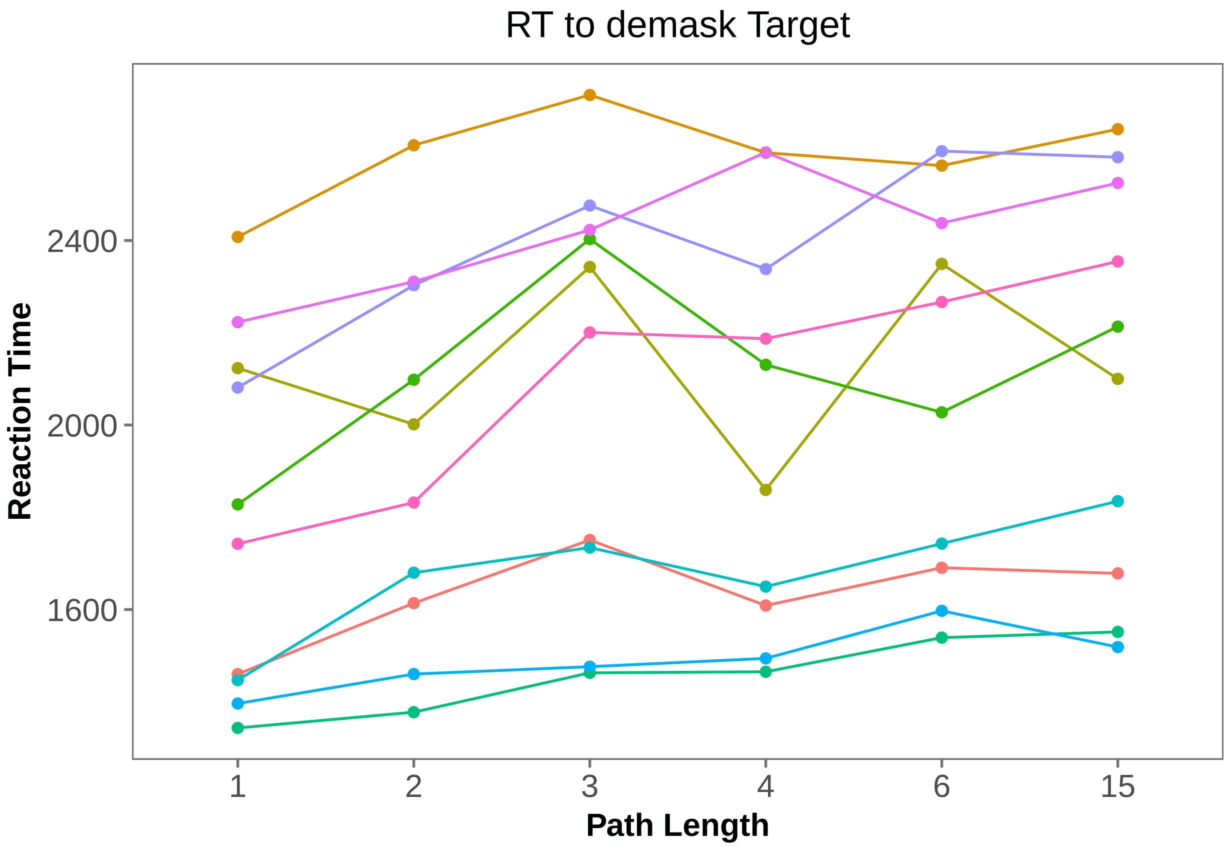
```
netdemask_rt_agg$pathlengthfac = ordered(as.factor(as.character(netdemask_rt_agg$pathlength)),

library(ggplot2)
library(ggthemes)
```

```r
netdemask_rt_agg %>%
  ggplot(aes(x = pathlengthfac, y = RTRecogniseTarget, group = 1))+
  geom_point()+
  geom_line(color = "red")+
   geom_errorbar(aes(ymin=RTRecogniseTarget - se, ymax=RTRecogniseTarget + se),
             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 to Recognise Target by Path Length")  +
   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

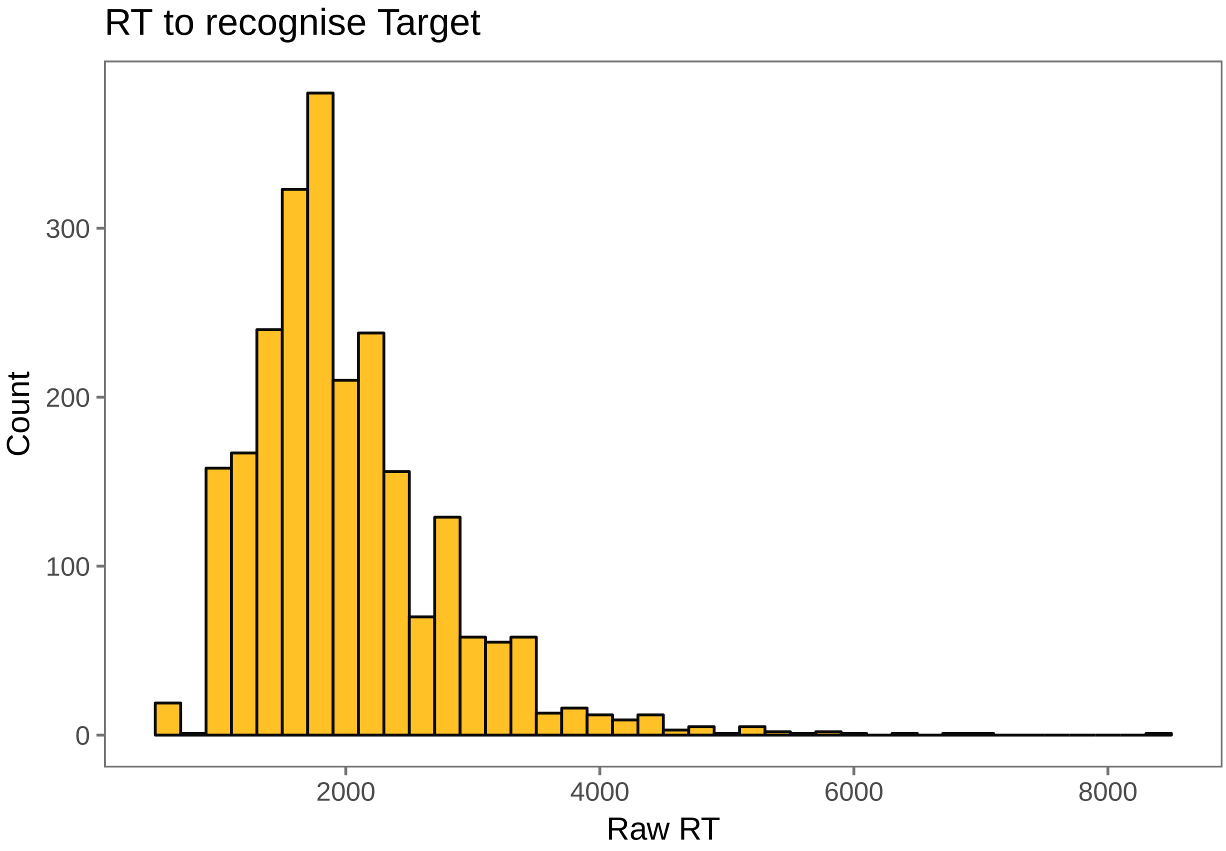
```r
library(ggplot2)
library(ggthemes)

netdemask_rt %>%
  ggplot(aes(x = pathlengthfac, y = RTRecogniseTarget,
             group = subject, color = subject))+
  geom_point()+
  geom_line()+
 theme_few()+
  guides(color = FALSE)+
 # scale_x_continuous(breaks = c(1,2,3,4,6,15))+
    xlab("Path Length") + ylab("Reaction Time") +
  ggtitle("RT to demask Target")  +
   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

Histogram of RT

```
library(ggplot2)
library(ggthemes)
ggplot(netdemask, aes(x = RTRecogniseTarget))+
geom_histogram(binwidth = 200, color = "gray4", fill = "goldenrod1")+
    theme_few()+
  #facet_wrap(~subject)+
  xlab("Raw RT") + ylab("Count") +
  ggtitle("RT to recognise Target")
```

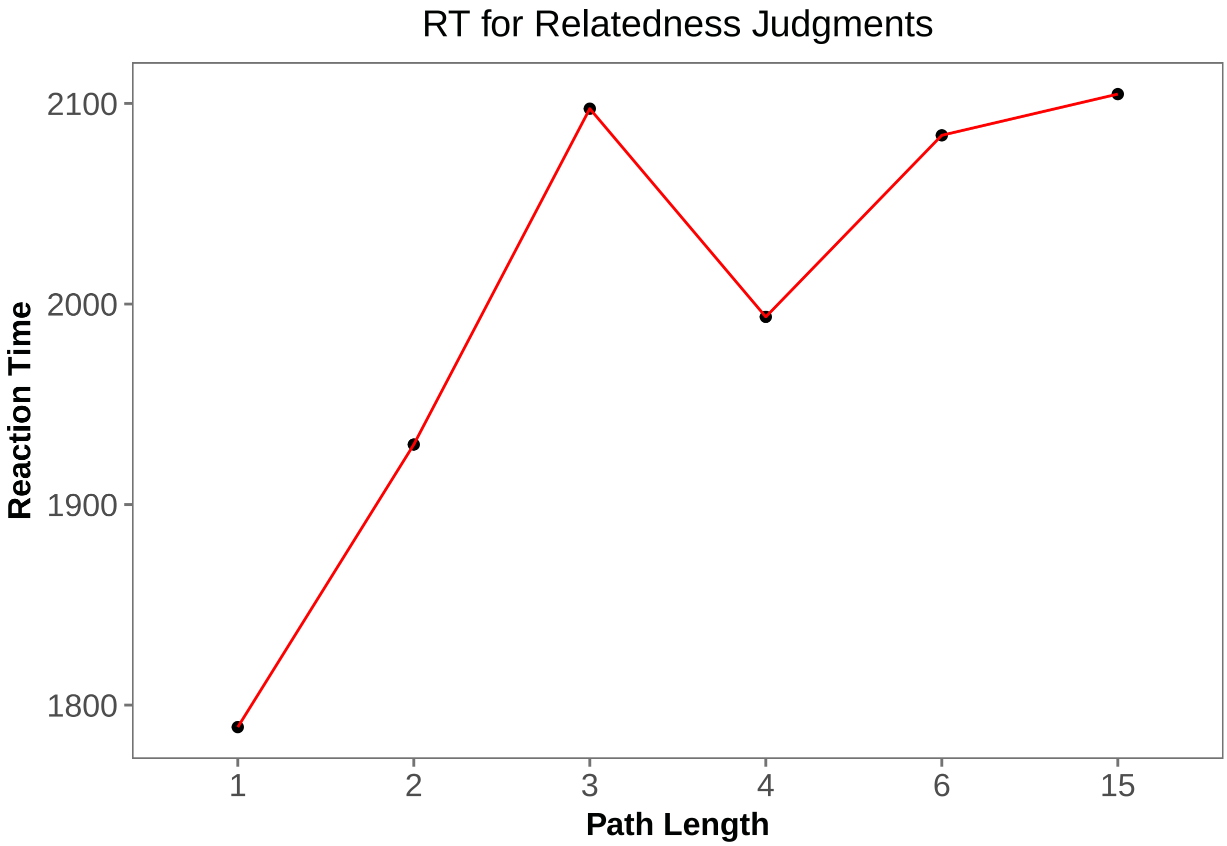


First Trim

```
library(dplyr)
netdemask_firsttrim = netdemask %>% filter(RTRecogniseTarget > 250 &
                                             RTRecogniseTarget < 7000)
```

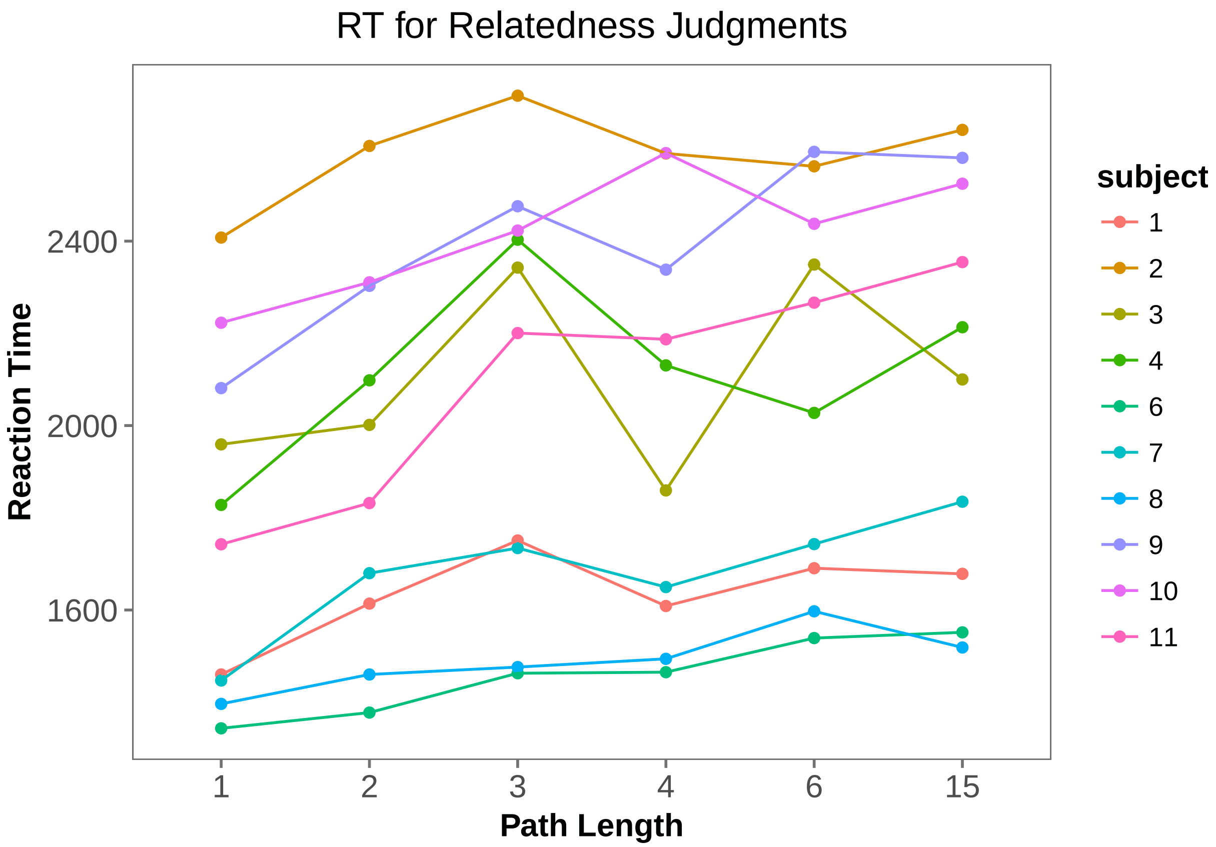
Raw RT aggregates After Trimming

```
netdemask_rt_firsttrim = group_by(netdemask_firsttrim, subject, pathlength ) %>%
  summarise_at(vars(RTRecogniseTarget), mean)

netdemask_rt_agg_firsttrim = group_by(netdemask_firsttrim, pathlength ) %>%
  summarise_at(vars(RTRecogniseTarget), mean)

netdemask_rt_agg_firsttrim$pathlengthfac = ordered(as.factor(as.character(netdemask_rt_agg_firsttrim$pa

library(ggplot2)
library(ggthemes)

netdemask_rt_agg_firsttrim %>%
  ggplot(aes(x = pathlengthfac, y = RTRecogniseTarget, 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)))
```



Subject Raw RT again

```
library(ggplot2)
library(ggthemes)

netdemask_rt_firsttrim$pathlengthfac = ordered(as.factor(as.character(netdemask_rt_firsttrim$pathlength
                            levels = c("1", "2", "3", "4", "6", "15"))
netdemask_rt_firsttrim$subject = as.factor(netdemask_rt_firsttrim$subject)
netdemask_rt_firsttrim %>%
  ggplot(aes(x = pathlengthfac, y = RTRecogniseTarget,
             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(1,2,3,4,5,6,10,15,20))+
    xlab("Path Length") + ylab("Reaction Time") +
  ggtitle("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)))
```



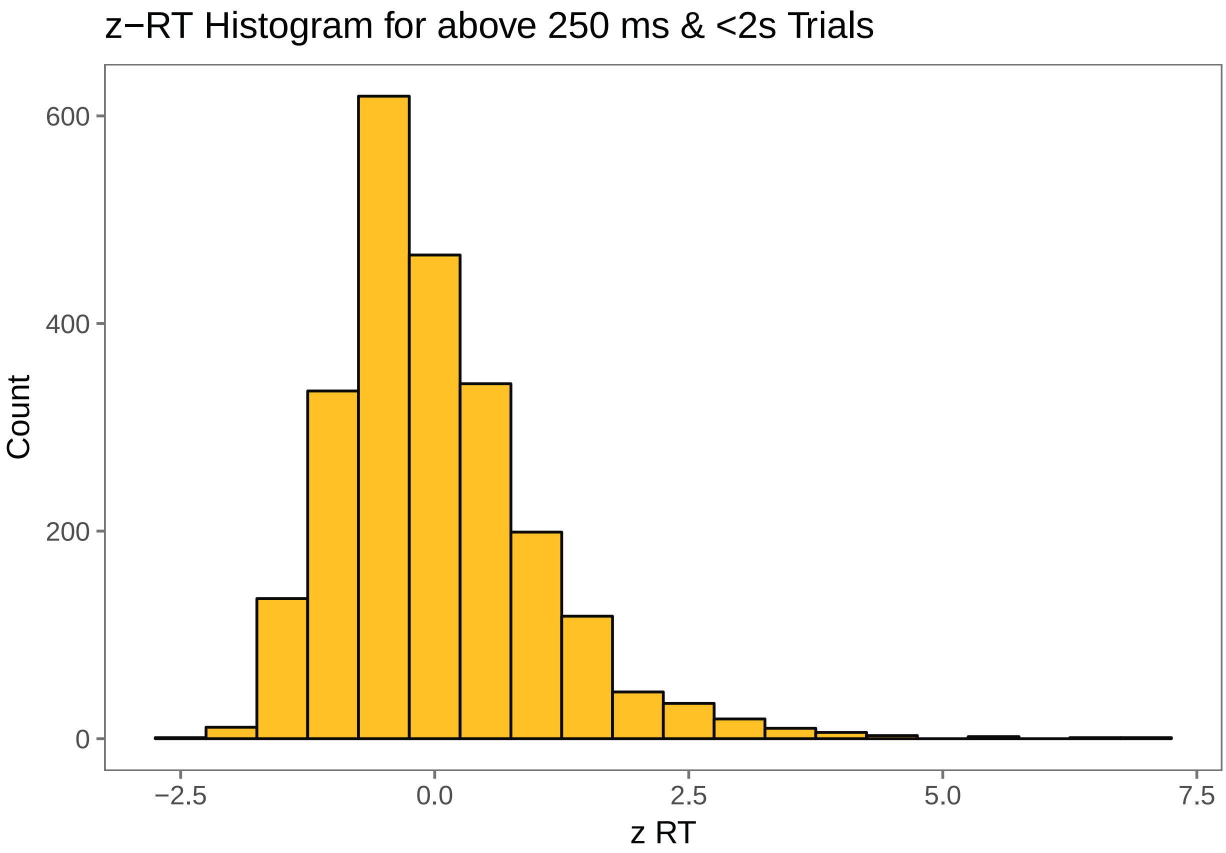
Making the z-scores

```
## aggregate per subject all IVs and DVs
meanRT = group_by(netdemask_firsttrim, subject) %>%
  summarise_at(vars(RTRecogniseTarget), mean)
colnames(meanRT) = c("subject", "MeanRTTarget")
```

```r
sdRT = group_by(netdemask_firsttrim, subject) %>%
  summarise_at(vars(RTRecogniseTarget), sd)
colnames(sdRT) = c("subject", "sdRTTarget")

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

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

## person and grand-mean centered scores using original and aggregate
library(dplyr)
netdemask_z = netdemask_z %>% mutate(zRTTarget =
                            (RTRecogniseTarget - MeanRTTarget)/sdRTTarget)

## checking: subject level means should be zero

sub_pic = group_by(netdemask_z, subject) %>%
  summarise_at(vars(zRTTarget), mean)
```

z-RT Distribution

```r
ggplot(netdemask_z, aes(x = zRTTarget))+
geom_histogram(binwidth = 0.5, color = "gray4", fill = "goldenrod1")+
    theme_few()+
  xlab("z RT") + ylab("Count") +
  ggtitle("z-RT Histogram for above 250 ms & <2s Trials")
```



Trimming z-RT

```
## trimming separately for prime and target
netdemask_z_trimmed_target = subset(netdemask_z, netdemask_z$zRTTarget < 3 &
                               netdemask_z$zRTTarget > -3)
```

Repeating z-scoring

```
library(dplyr)
## FOR TARGET
## aggregate per subject all IVs and DVs
meanRT_trim_target = group_by(netdemask_z_trimmed_target, subject) %>%
  summarise_at(vars(RTRecogniseTarget), mean)
colnames(meanRT_trim_target) = c("subject", "MeanRT_trim_target")

sdRT_trim_target = group_by(netdemask_z_trimmed_target, subject) %>%
  summarise_at(vars(RTRecogniseTarget), sd)
colnames(sdRT_trim_target) = c("subject", "sdRT_trim_target")

RT_agg_trim_target = merge(meanRT_trim_target, sdRT_trim_target, by = "subject")

## merge aggregate info with long data
new_netdemask_z_target = merge(netdemask_z_trimmed_target,
                        RT_agg_trim_target, by = "subject", all.x = T)

## person and grand-mean centered scores using original and aggregate
library(dplyr)
new_netdemask_z_target = new_netdemask_z_target %>%
  mutate(zRTTarget_trim = (RTRecogniseTarget - MeanRT_trim_target)/sdRT_trim_target)

## checking: subject level means should be zero

sub_pic = group_by(new_netdemask_z_target, subject) %>%
  summarise_at(vars(zRTTarget_trim), mean)


new_netdemask_z = new_netdemask_z_target
```

Aggregating zRT

```
z_netdemask_rt = group_by(new_netdemask_z, subject, pathlength ) %>%
  summarise_at(vars(zRTTarget_trim), mean)

z_rmisc = Rmisc::summarySE(new_netdemask_z,
                        measurevar = "zRTTarget_trim",
                        groupvars = c("pathlength"))
```

ANOVA

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

z_rt_aov = aov(data = z_netdemask_rt, zRTTarget_trim ~ pathlengthfac +
                     Error(subject/(pathlengthfac)))
summary(z_rt_aov)
```

```
##
## Error: subject
##           Df    Sum Sq   Mean Sq F value Pr(>F)
## Residuals  9 0.0003626 4.029e-05
##
## Error: subject:pathlengthfac
##                Df Sum Sq Mean Sq F value   Pr(>F)
## pathlengthfac  5  2.426  0.4852   16.76 2.51e-09 ***
## Residuals     45  1.303  0.0290
## ---
## 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)
```

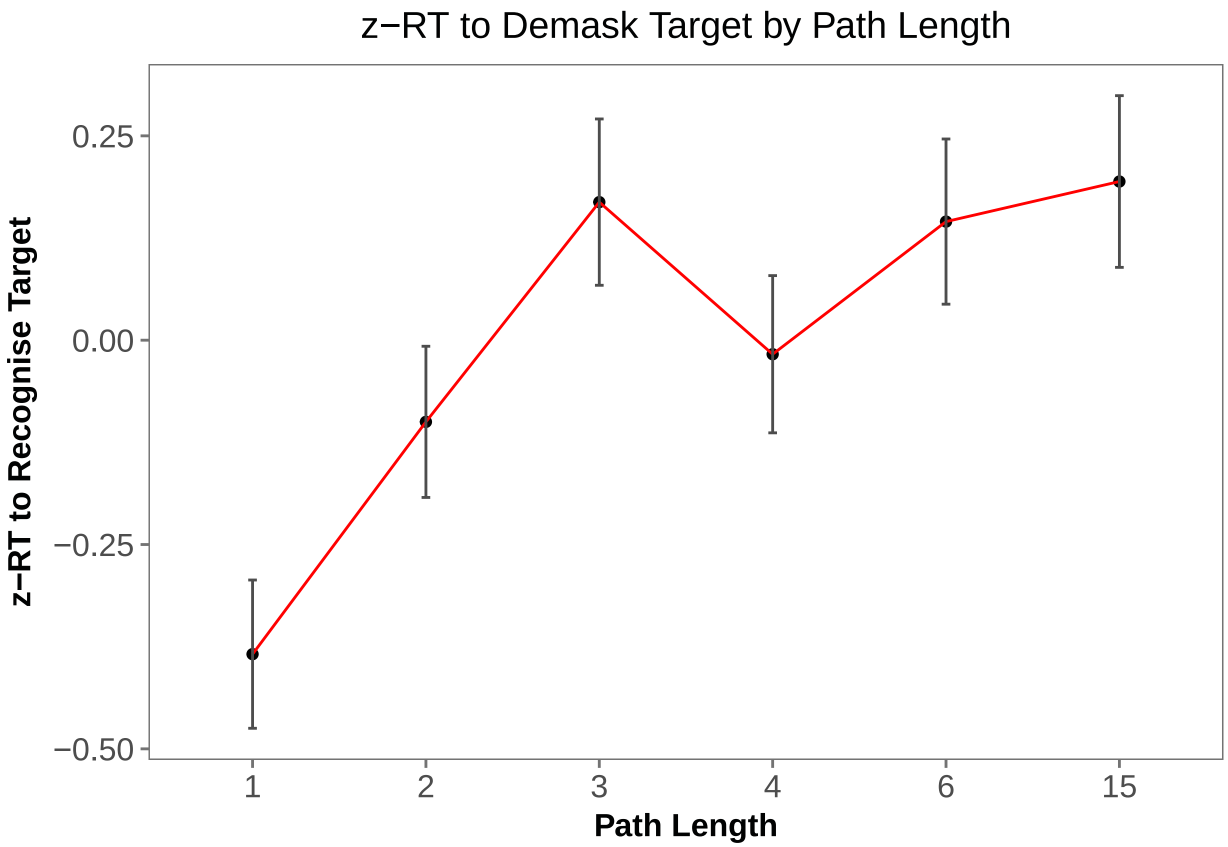
```
## Loading required package: mvtnorm

## 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(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
1	2 - 1	0.2809286	0.0760964	45	3.691745	0.0074327
2	4 - 1	0.3610196	0.0760964	45	4.744238	0.0002949
4	6 - 1	0.5268352	0.0760964	45	6.923256	0.0000002
5	6 - 2	0.2459065	0.0760964	45	3.231511	0.0262366
7	3 - 1	0.5492333	0.0760964	45	7.217595	0.0000001
8	3 - 2	0.2683047	0.0760964	45	3.525850	0.0118677
11	15 - 1	0.5745199	0.0760964	45	7.549892	0.0000000
12	15 - 2	0.2935913	0.0760964	45	3.858147	0.0045871
13	15 - 4	0.2135002	0.0760964	45	2.805654	0.0748917

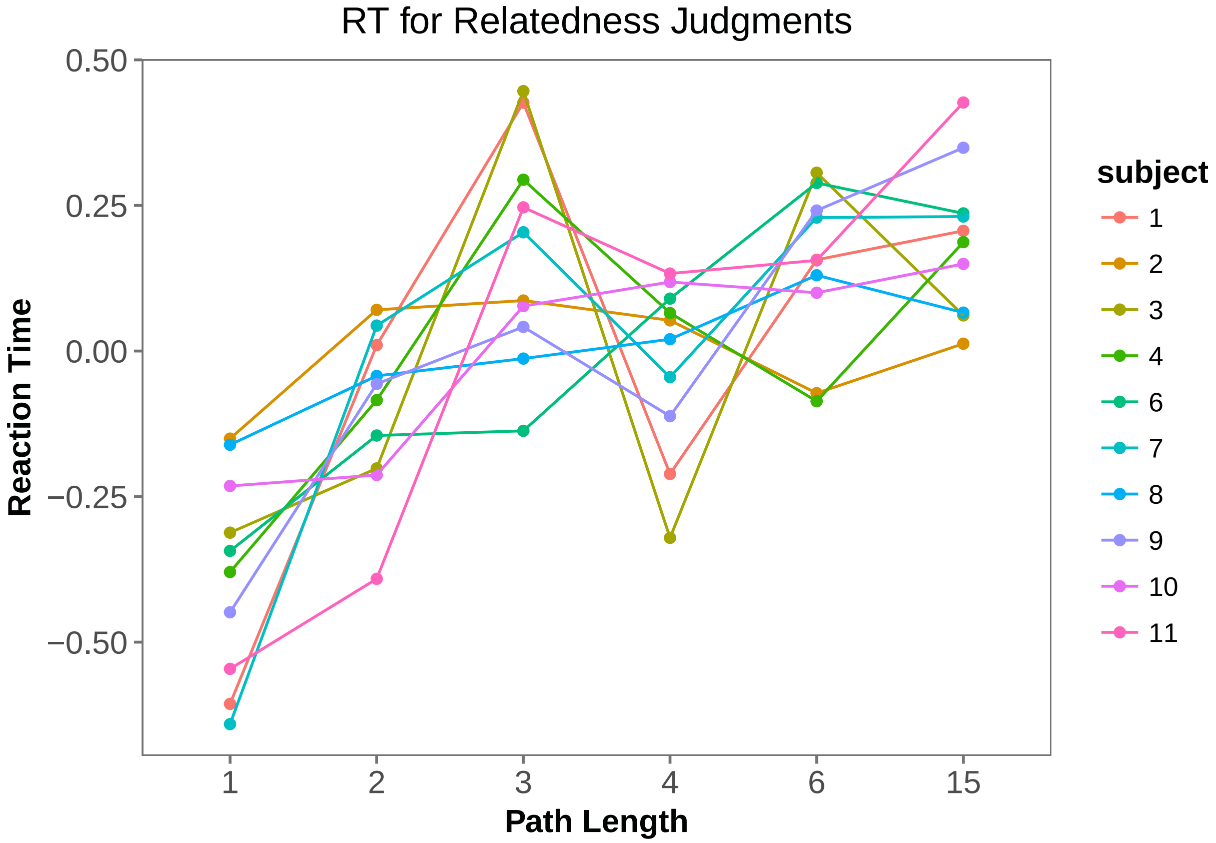
Plotting RTs: collapsed

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

z_rmisc$zRTTarget_trim = as.numeric(z_rmisc$zRTTarget_trim)

library(ggplot2)
library(ggthemes)

z_rmisc %>%
  ggplot(aes(x = pathlengthfac, y = zRTTarget_trim, group = 1))+
  geom_point()+
#  geom_smooth(method = "loess")+
geom_line(color = "red")+
   geom_errorbar(aes(ymin=zRTTarget_trim - ci, ymax=zRTTarget_trim + ci),
             width=.05, color = "gray30",
             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 to Recognise Target") +
  ggtitle("z-RT to Demask Target by Path Length")  +
   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 z RT

```
library(ggplot2)
library(ggthemes)


z_netdemask_rt %>%
  ggplot(aes(x = pathlengthfac, y = zRTTarget_trim,
             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(1,2,3,4,5,6,10,15,20))+
    xlab("Path Length") + ylab("Reaction Time") +
  ggtitle("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)))
```



Other Networks

Steyvers Non Directed

```
library(lme4)
```

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

```
new_netdemask_z$Undirected = as.double(as.character(new_netdemask_z$Undirected))
new_netdemask_z$Directed = as.double(as.character(new_netdemask_z$Directed))

new_netdemask_z$undirectedfac = ordered(as.factor(as.character(new_netdemask_z$Undirected)),


contrasts(new_netdemask_z$undirectedfac) = contr.treatment(4, base = 4)
RTprime_undirected = lmer(data = new_netdemask_z,
                          zRTTarget_trim ~ undirectedfac +
                            (1|subject) + (1|ItemNumber))
summary(RTprime_undirected)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: zRTTarget_trim ~ undirectedfac + (1 | subject) + (1 | ItemNumber)
##    Data: new_netdemask_z
##
## REML criterion at convergence: 6403.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.8398 -0.6646 -0.1247  0.5707  3.4958
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  ItemNumber (Intercept) 0.1114   0.3338
##  subject    (Intercept) 0.0000   0.0000
##  Residual               0.8515   0.9228
## Number of obs: 2315, groups:  ItemNumber, 240; subject, 10
##
## Fixed effects:
##                Estimate Std. Error t value
## (Intercept)      0.3497     0.1061   3.295
## undirectedfac1  -0.6986     0.1440  -4.850
## undirectedfac2  -0.4853     0.1154  -4.204
## undirectedfac3  -0.2030     0.1149  -1.767
##
## Correlation of Fixed Effects:
##             (Intr) undrc1 undrc2
## undirctdfc1 -0.737
## undirctdfc2 -0.920  0.678
## undirctdfc3 -0.924  0.681  0.850
```

```
car::Anova(RTprime_undirected)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: zRTTarget_trim
##                Chisq Df Pr(>Chisq)
## undirectedfac 43.781  3   1.68e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
RTprime_undirected_quad = lmer(data = new_netdemask_z,
                          zRTTarget_trim ~ Undirected +
                            I(Undirected^2)+
                            (1|subject) + (1|ItemNumber))
summary(RTprime_undirected_quad)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: zRTTarget_trim ~ Undirected + I(Undirected^2) + (1 | subject) +
##     (1 | ItemNumber)
##    Data: new_netdemask_z
##
## REML criterion at convergence: 6404.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.8326 -0.6660 -0.1242  0.5681  3.4883
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  ItemNumber (Intercept) 0.1109   0.3330
##  subject    (Intercept) 0.0000   0.0000
##  Residual               0.8515   0.9228
## Number of obs: 2315, groups:  ItemNumber, 240; subject, 10
##
## Fixed effects:
##                  Estimate Std. Error t value
## (Intercept)     -0.631306   0.237695  -2.656
## Undirected       0.259267   0.197205   1.315
## I(Undirected^2) -0.001658   0.039237  -0.042
##
## Correlation of Fixed Effects:
##             (Intr) Undrct
## Undirected  -0.969
## I(Undrct^2)  0.909 -0.981
```

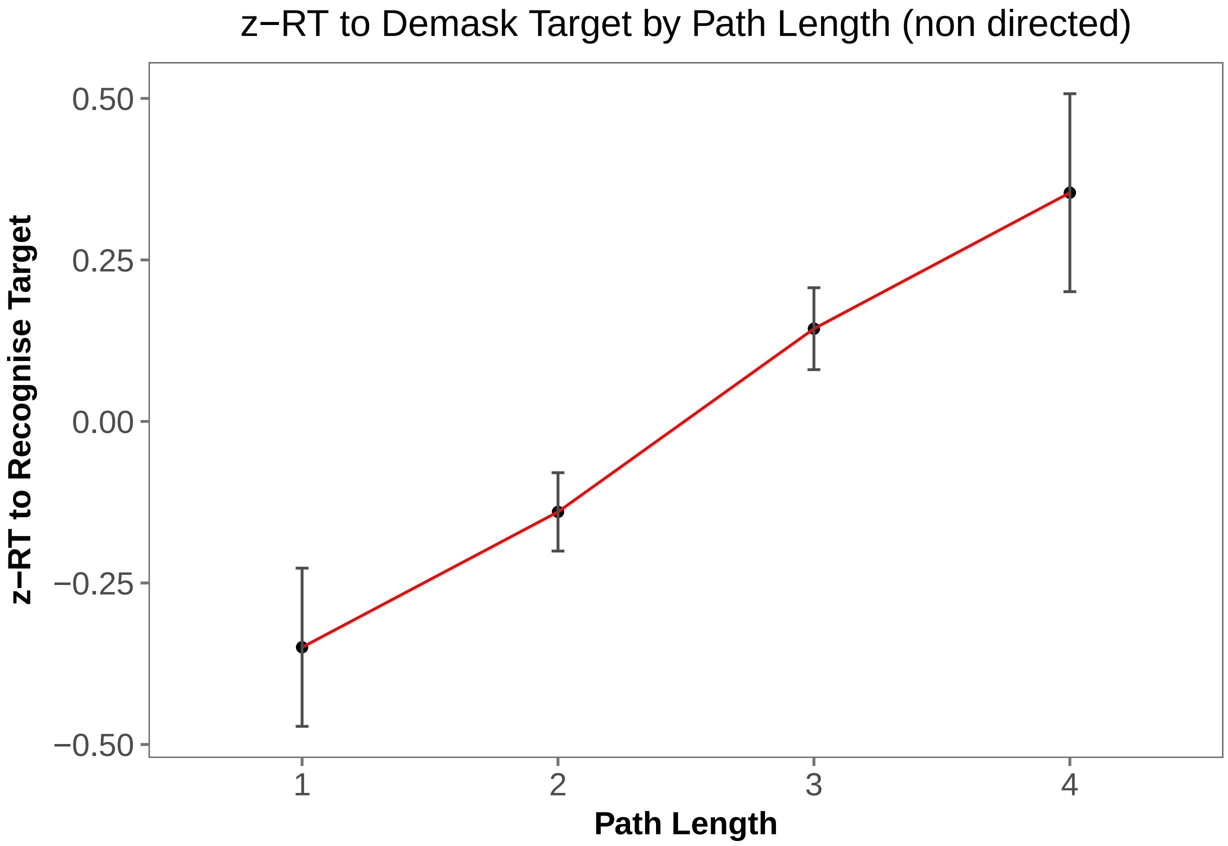
```
car::Anova(RTprime_undirected_quad)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: zRTTarget_trim
##                  Chisq Df Pr(>Chisq)
## Undirected      1.7285  1     0.1886
## I(Undirected^2) 0.0018  1     0.9663
```

Plot

```
z_rmisc_undirected = Rmisc::summarySE(new_netdemask_z,
                        measurevar = "zRTTarget_trim",
                        groupvars = c("Undirected"))
z_rmisc_undirected = z_rmisc_undirected %>% filter(Undirected != "NA")
z_rmisc_undirected$undirectedfac = ordered(as.factor(as.character(z_rmisc_undirected$Undirected)),

z_rmisc_undirected$zRTTarget_trim = as.numeric(z_rmisc_undirected$zRTTarget_trim)

library(ggplot2)
library(ggthemes)
```

```
z_rmisc_undirected %>%
  ggplot(aes(x = undirectedfac, y = zRTTarget_trim, group = 1))+
  geom_point()+
#  geom_smooth(method = "loess")+
geom_line(color = "red")+
   geom_errorbar(aes(ymin=zRTTarget_trim - ci, ymax=zRTTarget_trim + ci),
             width=.05, color = "gray30",
             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 to Recognise Target") +
  ggtitle("z-RT to Demask Target by Path Length (non directed)")  +
   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)))
```



Steyvers Directed

```
library(lme4)
new_netdemask_z$newdirected = ifelse(new_netdemask_z$Directed == "Inf" |
                              new_netdemask_z$Directed == "NA", NA,
                              new_netdemask_z$Directed)
```

```r
new_netdemask_z$directedcollapsed = ifelse((new_netdemask_z$newdirected == "5" |
                                      new_netdemask_z$newdirected == "6" |
                                       new_netdemask_z$newdirected == "7" |
                                       new_netdemask_z$newdirected == "8"), "H",
                              new_netdemask_z$newdirected)


new_netdemask_z$directedfac =
  ordered(as.factor(as.character(new_netdemask_z$newdirected)),
                            levels = c("1", "2", "3", "4", "5",
                                       "6", "7", "8"))
contrasts(new_netdemask_z$directedfac) = contr.treatment(8, base = 5)

new_netdemask_z$collapsedfac =
  ordered(as.factor(as.character(new_netdemask_z$directedcollapsed)),
                            levels = c("1", "2", "3", "4", "H"))
contrasts(new_netdemask_z$collapsedfac) = contr.treatment(5, base = 5)


RTprime_directed = lmer(data = new_netdemask_z,
                          zRTTarget_trim ~ directedfac +
                            (1|subject) + (1|ItemNumber))
summary(RTprime_directed)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: zRTTarget_trim ~ directedfac + (1 | subject) + (1 | ItemNumber)
##    Data: new_netdemask_z
##
## REML criterion at convergence: 6172.4
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.8081 -0.6862 -0.1310  0.5780  3.6486
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  ItemNumber (Intercept) 0.08422  0.2902
##  subject    (Intercept) 0.00000  0.0000
##  Residual               0.84442  0.9189
## Number of obs: 2248, groups:  ItemNumber, 240; subject, 10
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)   0.14887    0.05472   2.720
## directedfac1 -0.44175    0.11350  -3.892
## directedfac2 -0.42081    0.07771  -5.415
## directedfac3 -0.24945    0.07188  -3.470
## directedfac4 -0.11192    0.06648  -1.684
## directedfac6  0.14481    0.10320   1.403
## directedfac7  0.69087    0.25641   2.694
## directedfac8  0.44442    0.33997   1.307
##
## Correlation of Fixed Effects:
```

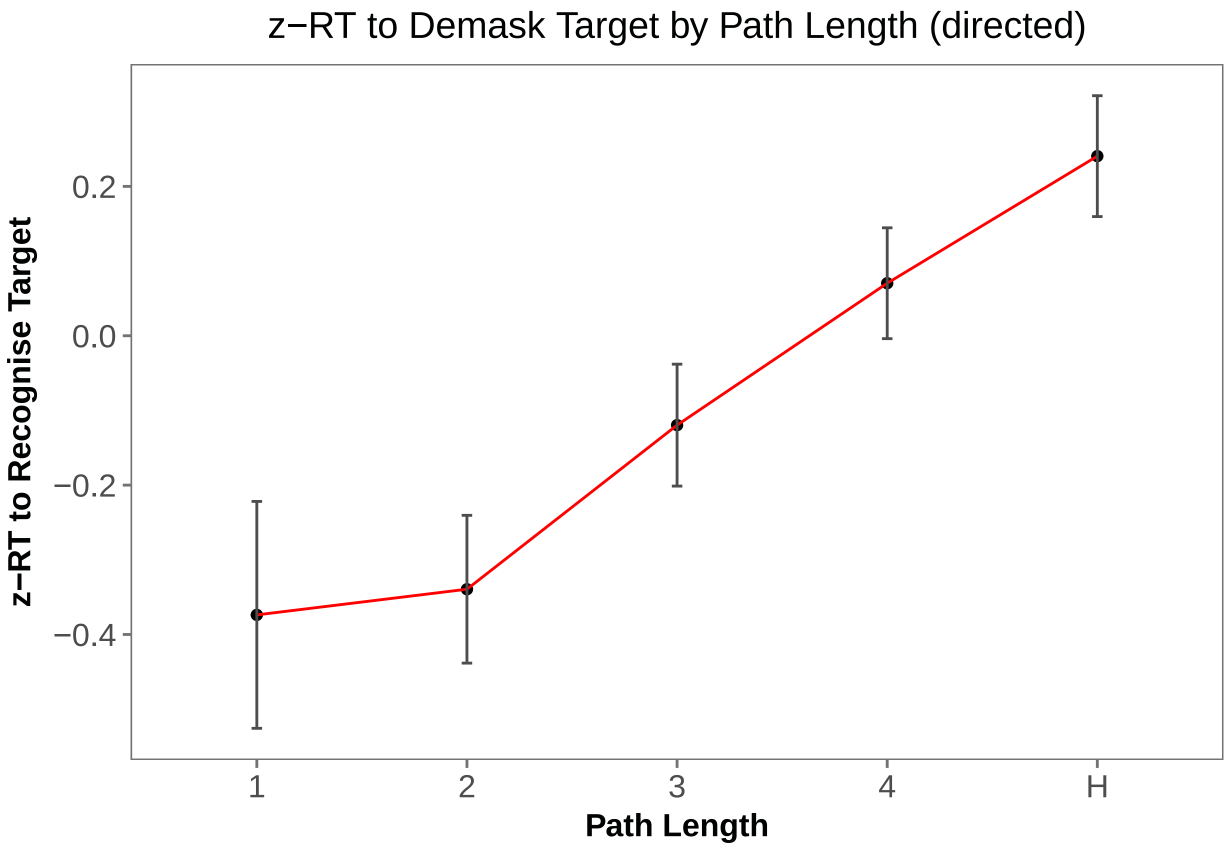
```
##             (Intr) drctd1 drctd2 drctd3 drctd4 drctd6 drctd7
## directedfc1 -0.475
## directedfc2 -0.662  0.350
## directedfc3 -0.704  0.352  0.507
## directedfc4 -0.756  0.379  0.527  0.576
## directedfc6 -0.464  0.234  0.341  0.336  0.383
## directedfc7 -0.209  0.101  0.140  0.150  0.186  0.099
## directedfc8 -0.157  0.077  0.127  0.130  0.121  0.075  0.033
```

```
car::Anova(RTprime_directed)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: zRTTarget_trim
##              Chisq Df Pr(>Chisq)
## directedfac 67.282  7  5.223e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

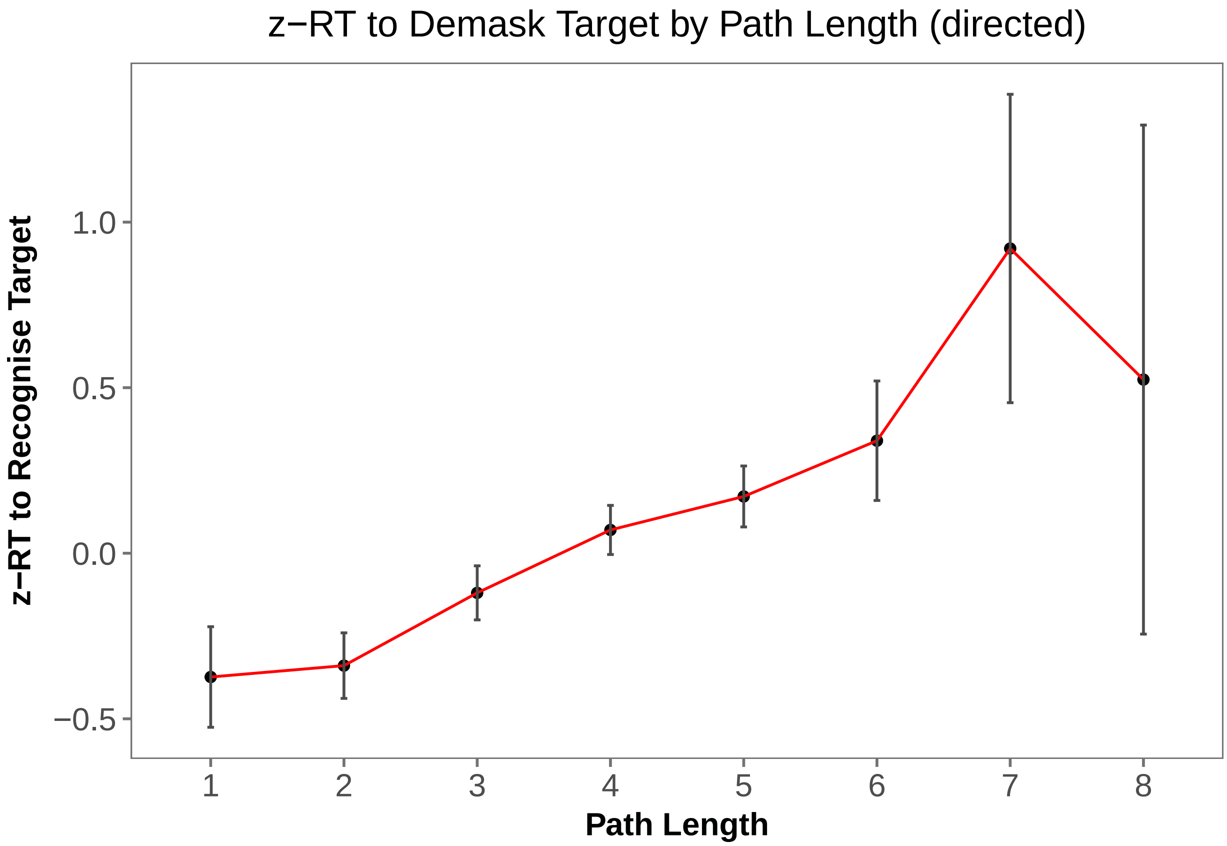
Plot Collapsed

```
z_rmisc_directed = Rmisc::summarySE(new_netdemask_z,
                        measurevar = "zRTTarget_trim",
                        groupvars = c("collapsedfac"))
z_rmisc_directed = z_rmisc_directed %>% filter(collapsedfac != "NA")
z_rmisc_directed$collapsedfac2 = ordered(as.factor(as.character(z_rmisc_directed$collapsedfac)),

z_rmisc_directed$zRTTarget_trim = as.numeric(z_rmisc_directed$zRTTarget_trim)

library(ggplot2)
library(ggthemes)

z_rmisc_directed %>%
  ggplot(aes(x = collapsedfac2, y = zRTTarget_trim, group = 1))+
  geom_point()+
#  geom_smooth(method = "loess")+
geom_line(color = "red")+
   geom_errorbar(aes(ymin=zRTTarget_trim - ci, ymax=zRTTarget_trim + ci),
             width=.05, color = "gray30",
             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 to Recognise Target") +
  ggtitle("z-RT to Demask Target by Path Length (directed)")  +
   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)))
```



Plot Not Collapsed

```
z_rmisc_directed = Rmisc::summarySE(new_netdemask_z,
                        measurevar = "zRTTarget_trim",
                        groupvars = c("directedfac"))
z_rmisc_directed = z_rmisc_directed %>% filter(directedfac != "NA")
z_rmisc_directed$collapsedfac2 = ordered(as.factor(as.character(z_rmisc_directed$directedfac)),

z_rmisc_directed$zRTTarget_trim = as.numeric(z_rmisc_directed$zRTTarget_trim)

library(ggplot2)
library(ggthemes)

z_rmisc_directed %>%
  ggplot(aes(x = collapsedfac2, y = zRTTarget_trim, group = 1))+
  geom_point()+
#  geom_smooth(method = "loess")+
geom_line(color = "red")+
   geom_errorbar(aes(ymin=zRTTarget_trim - ci, ymax=zRTTarget_trim + ci),
             width=.05, color = "gray30",
             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 to Recognise Target") +
  ggtitle("z-RT to Demask Target by Path Length (directed)")  +
   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)))
```


z−RT to Demask Target by Path Length (directed)
z−RT to Recognise Target
Path Length