# Network Demasking

## Reading the Data

```r
library(dplyr)
netdemask = read.csv("Compiled_NetworksDemasking.csv", header = TRUE, sep = ",")
```

## Raw Reaction Time

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

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

### ANOVA

```r
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  3 1994601  664867
##
## Error: subject:pathlengthfac
##               Df Sum Sq Mean Sq F value Pr(>F)
## pathlengthfac  5  88943   17789   1.817   0.17
## Residuals     15 146835    9789
```

### Plotting RTs

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

library(ggplot2)
library(ggthemes)

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)))
```



RT to Recognise Target by Path Length
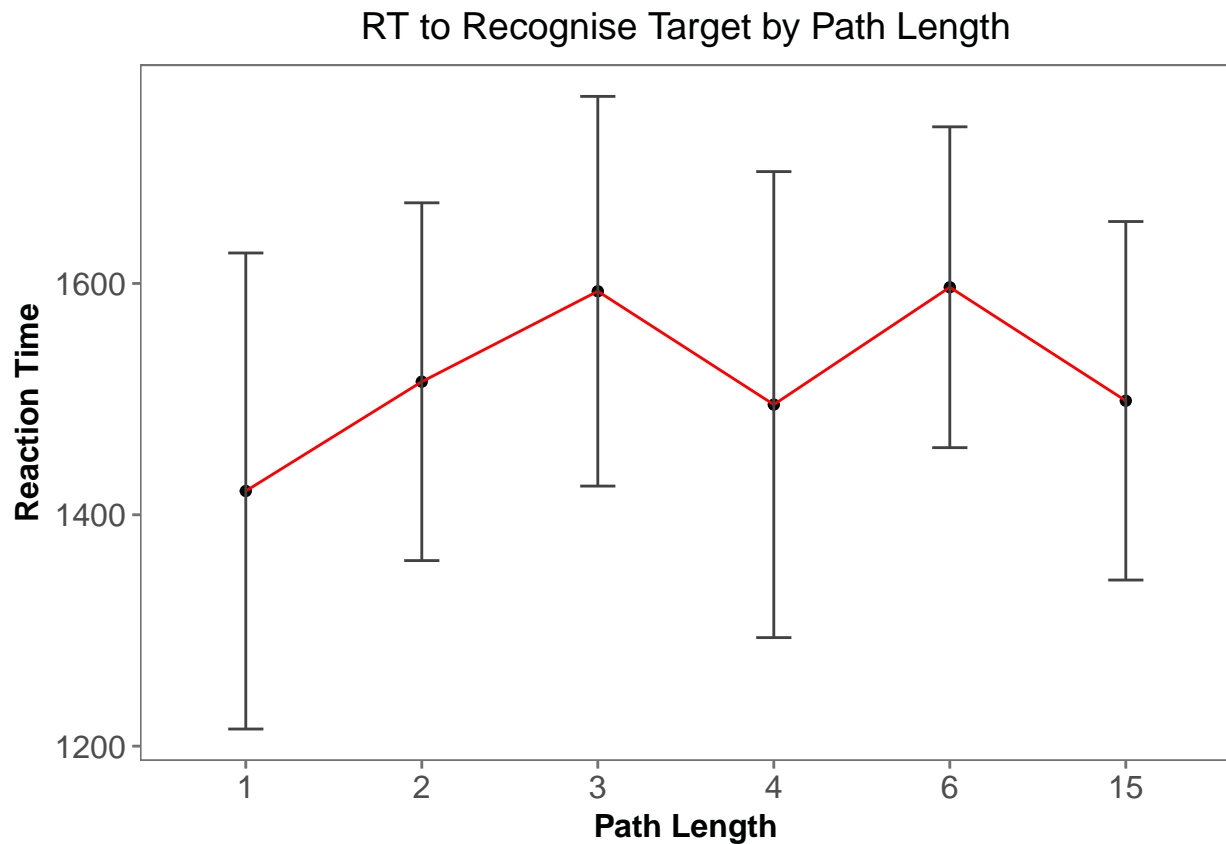
**Subject-Wise**

```
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)))
```
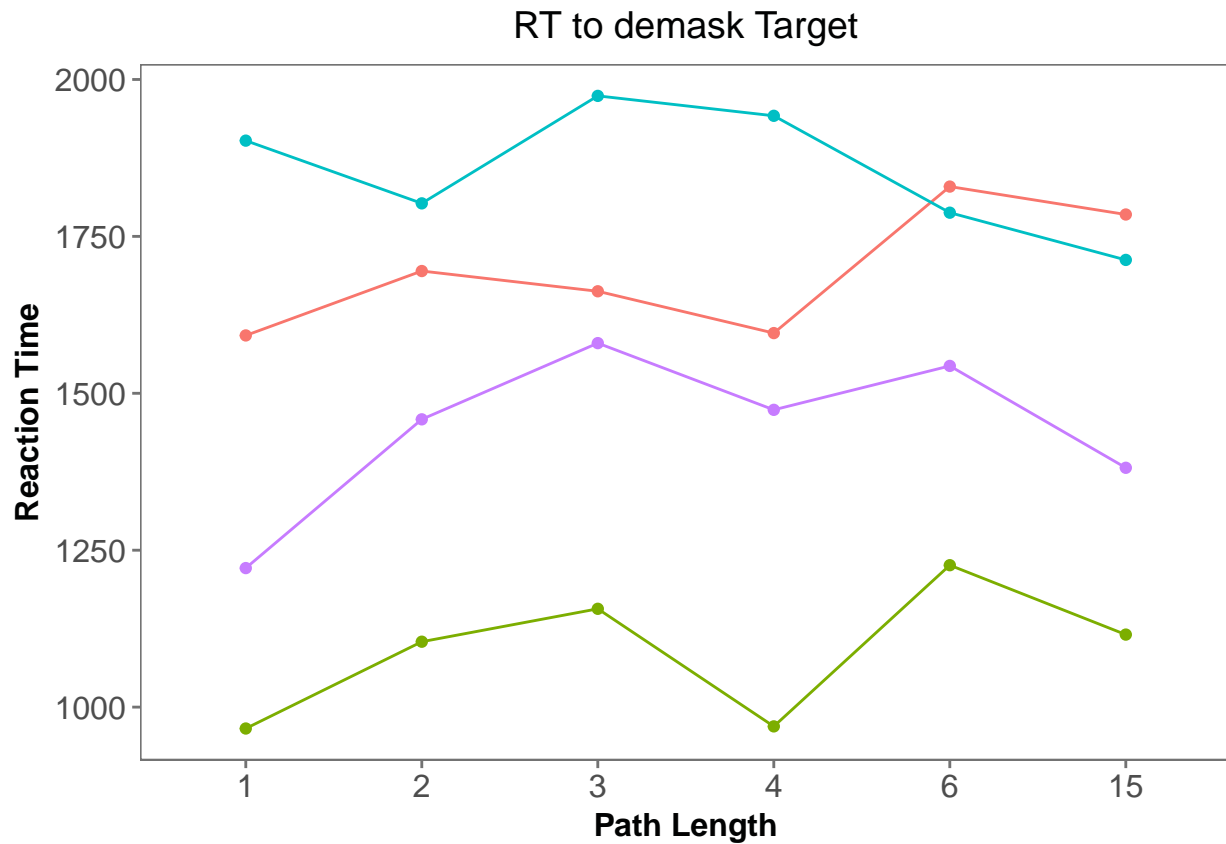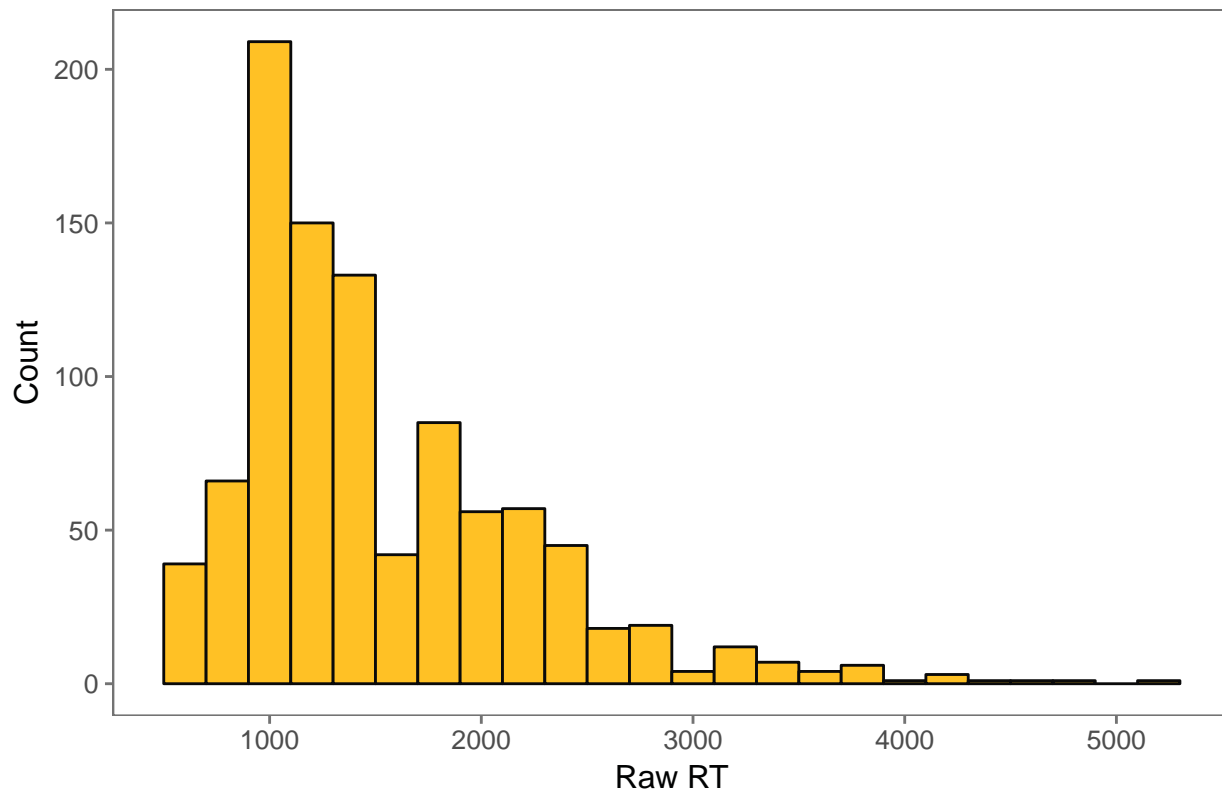


## z-scored Reaction Time

### 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")
```

## RT to recognise Target



### First Trim

```
library(dplyr)
netdemask_firsttrim = netdemask
```

### Raw RT aggregates After Trimming

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

netdemask_rt_agg_firsttrim = group_by(netdemask_firsttrim, pathlength ) %>%
  summarise_at(vars(RTRecognisePrime,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),
```
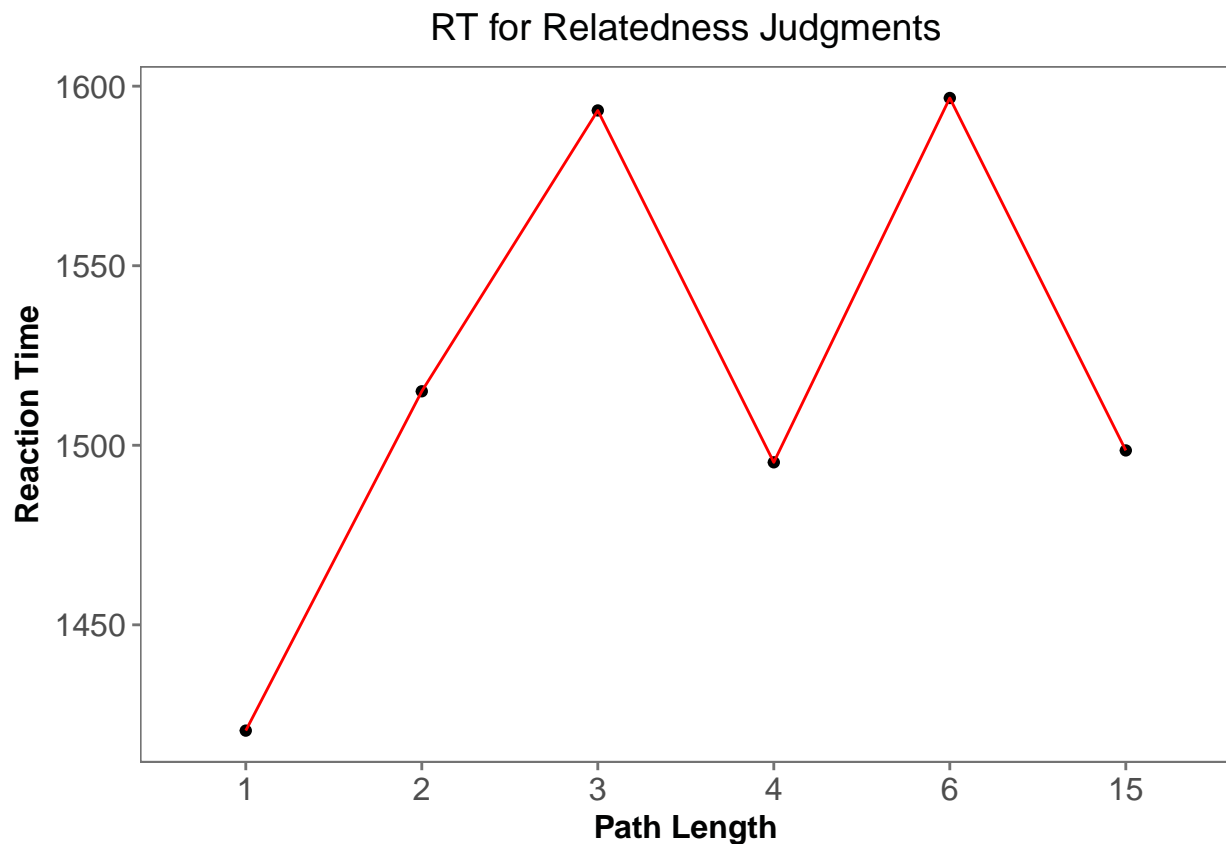
```
#             width=.2, color = "gray26",
 #             position = position_dodge(0.7))+
theme_few()+
 #scale_x_continuous(breaks = c(1,2,3,4,6,15))+
   xlab("Path Length") + ylab("Reaction Time") +
 ggtitle("RT for Relatedness Judgments")  +
  theme(axis.text = element_text(size = rel(1)),
        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)))
```



RT for Relatedness Judgments
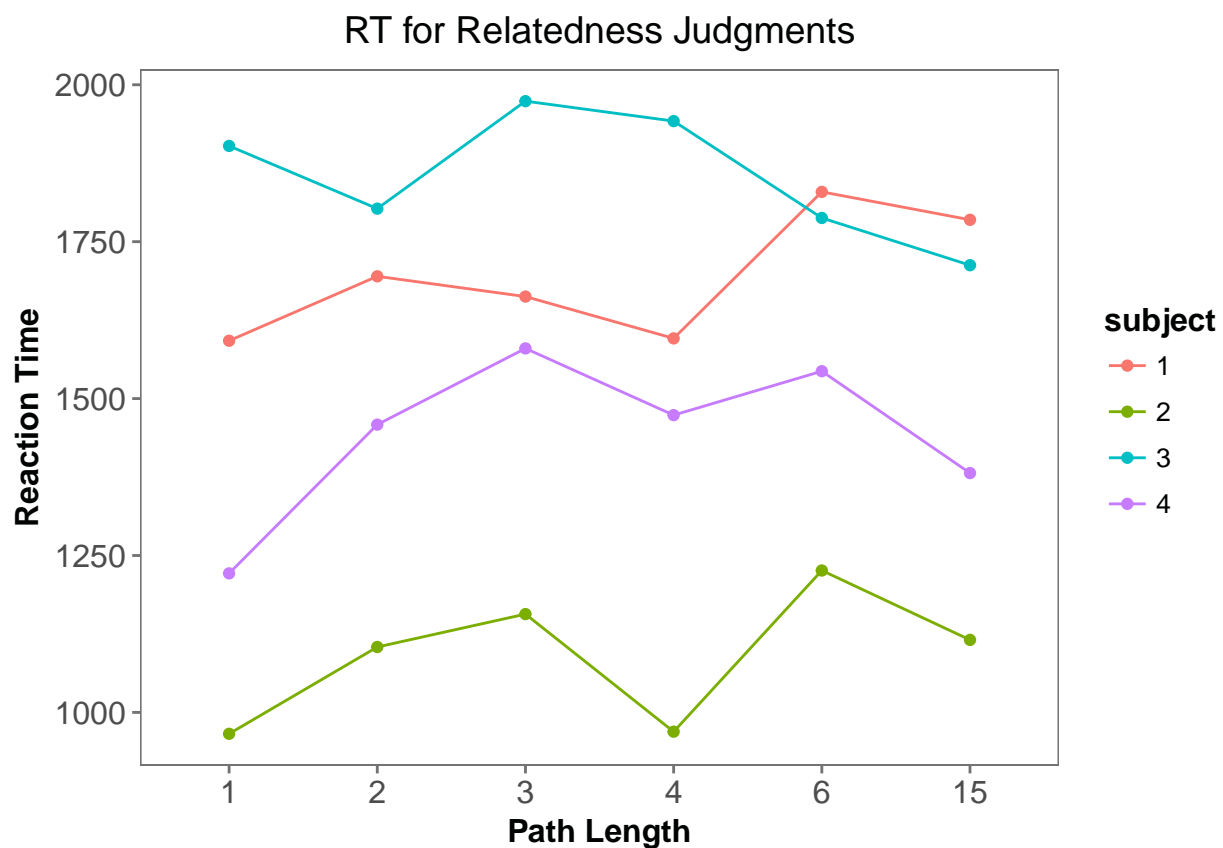
### Subject Raw RT again

```
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(RTRecognisePrime,RTRecogniseTarget), mean)
colnames(meanRT) = c("subject", "MeanRTPrime", "MeanRTTarget")

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

```r
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,
                    zRTPrime =  (RTRecognisePrime - MeanRTPrime)/sdRTPrime)

## checking: subject level means should be zero

sub_pic = group_by(netdemask_z, subject) %>%
  summarise_at(vars(zRTTarget, zRTPrime), mean)
```
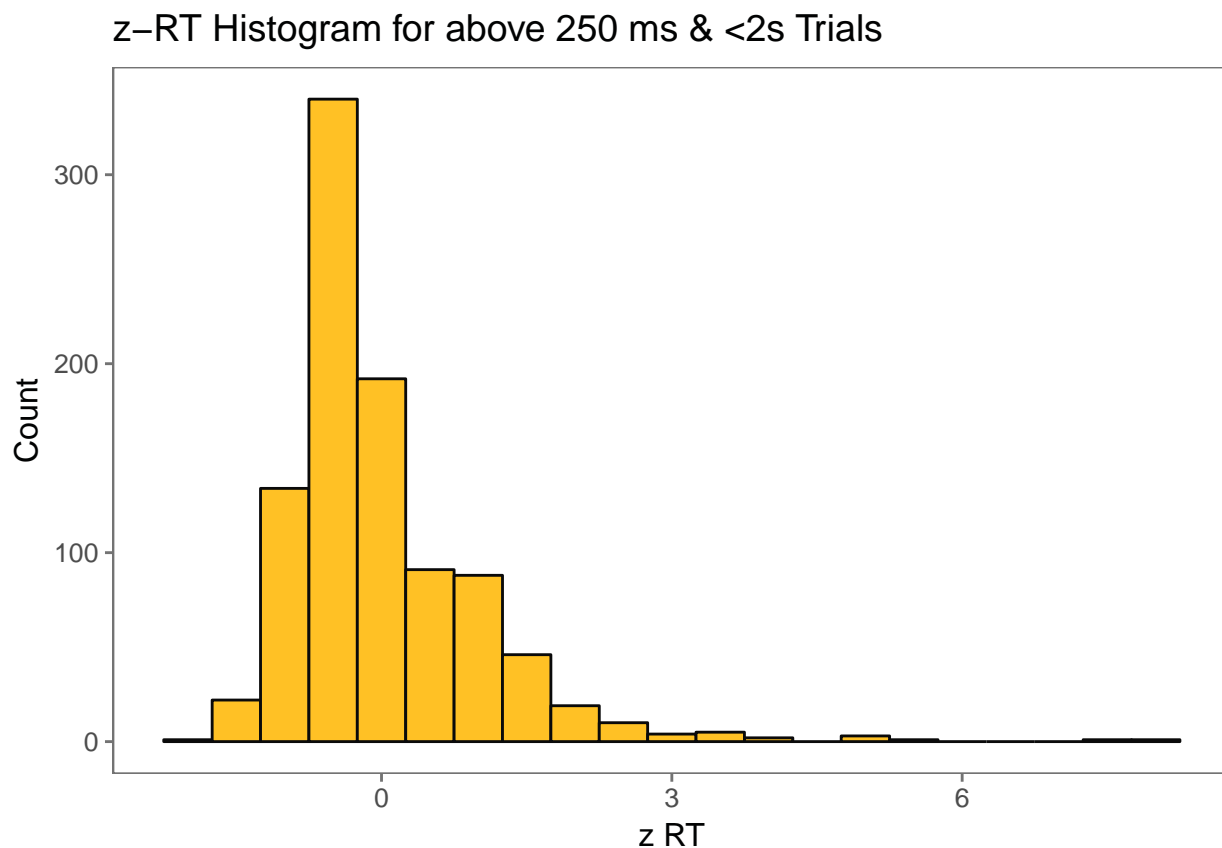
## z-RT Distribution

```r
ggplot(netdemask_z, aes(x = zRTPrime))+
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")
```



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)

netdemask_z_trimmed_prime = subset(netdemask_z, netdemask_z$zRTPrime < 3 &
                                   netdemask_z$zRTPrime > -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)

## FOR PRIME


meanRT_trim_prime = group_by(netdemask_z_trimmed_prime, subject) %>%
  summarise_at(vars(RTRecognisePrime), mean)
colnames(meanRT_trim_prime) = c("subject", "MeanRT_trim_prime")

sdRT_trim_prime = group_by(netdemask_z_trimmed_prime, subject) %>%
  summarise_at(vars(RTRecognisePrime), sd)
colnames(sdRT_trim_prime) = c("subject", "sdRT_trim_prime")

RT_agg_trim_prime = merge(meanRT_trim_prime, sdRT_trim_prime, by = "subject")

## merge aggregate info with long data
```

```r
new_netdemask_z_prime = merge(netdemask_z_trimmed_prime,
                              RT_agg_trim_prime, by = "subject", all.x = T)

## person and grand-mean centered scores using original and aggregate
library(dplyr)
new_netdemask_z_prime = new_netdemask_z_prime %>%
  mutate(zRTPrime_trim = (RTRecognisePrime - MeanRT_trim_prime)/sdRT_trim_prime)

## checking: subject level means should be zero

sub_pic = group_by(new_netdemask_z_prime, subject) %>%
  summarise_at(vars(zRTPrime_trim), mean)

## now we have separately z-scored RTprime and RTtarget. Need to combine.
## taking only necessary columns
new_netdemask_z_prime = new_netdemask_z_prime[,c(1,5,40)]

new_netdemask_z = merge(new_netdemask_z_target,
                        new_netdemask_z_prime,
                        by  = c("subject", "Trial"))
```

## Aggregating zRT

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

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

## ANOVA

```r
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   3 0.0009301 0.00031
##
## Error: subject:pathlengthfac
##               Df Sum Sq Mean Sq F value Pr(>F)
## pathlengthfac  5 0.2597 0.05194   1.385  0.285
## Residuals     15 0.5624 0.03749
```

```
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 ——— ——— — — ——— ———

## 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)))
```

## z–RT to Demask Target by Path Length
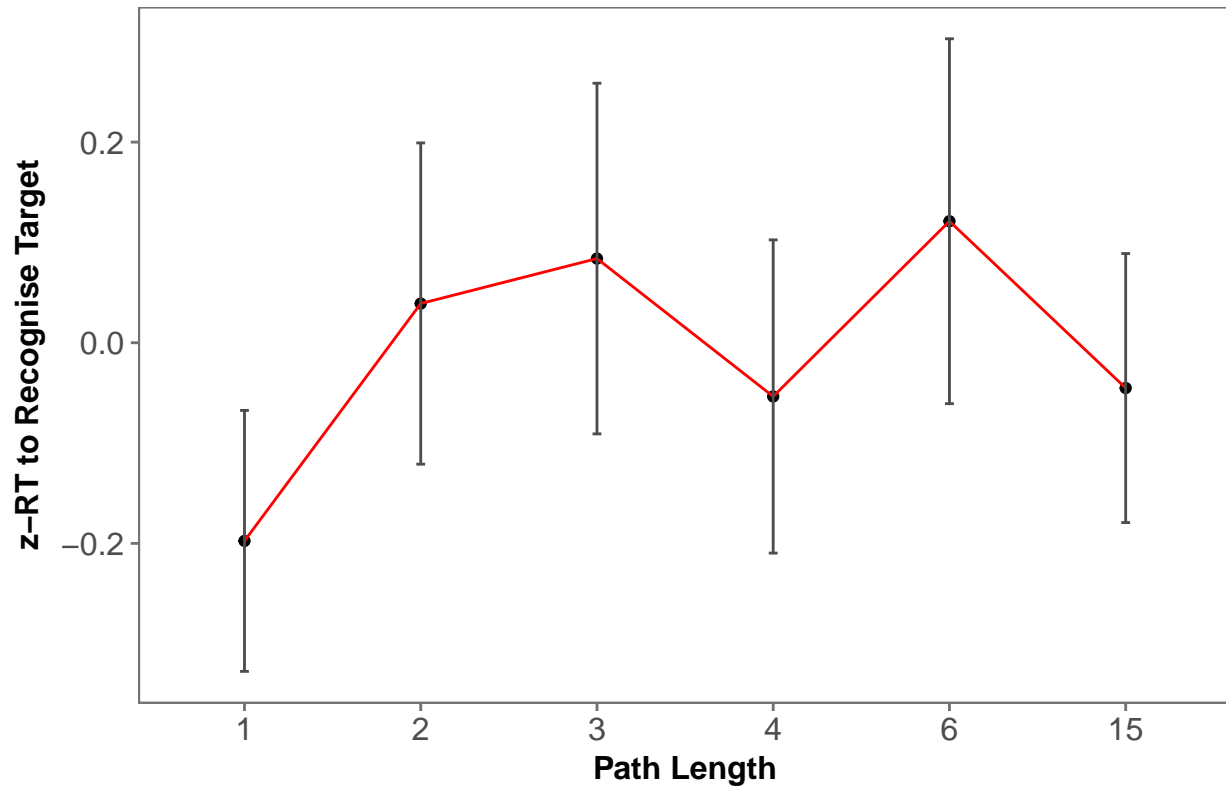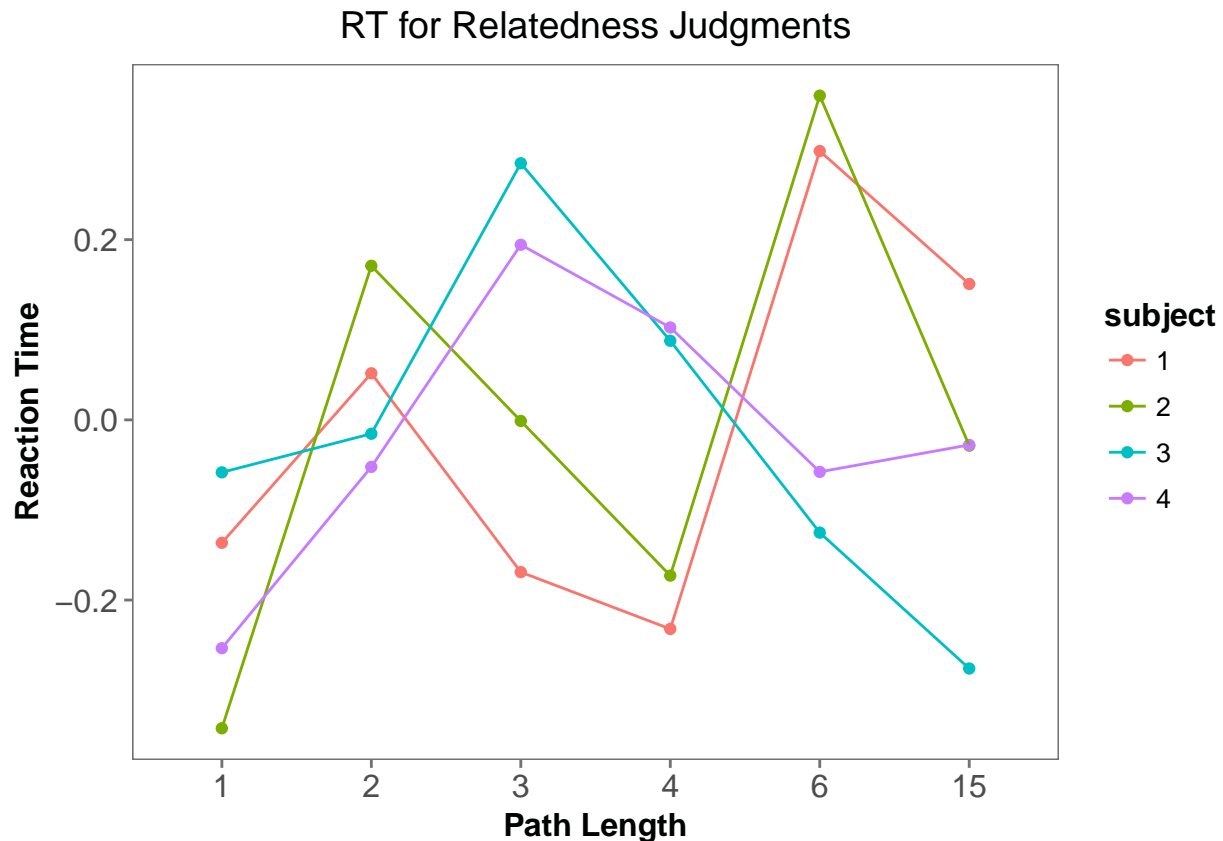


## Subject z RT

```
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)))
```



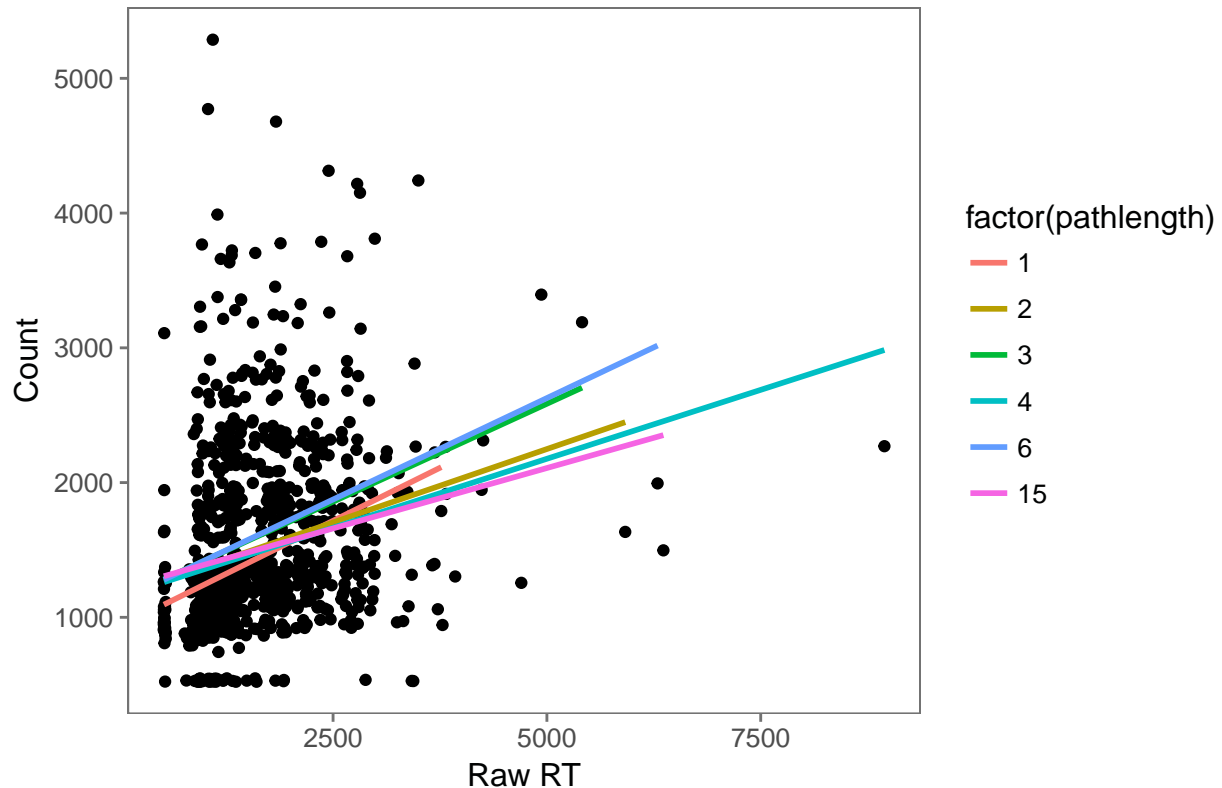## Effect of Prime on Target

**Simple Scatter Plot**

```
ggplot(netdemask, aes(x = RTRecognisePrime, y = RTRecogniseTarget))+
geom_point()+
  geom_smooth(method = "lm", aes(group = factor(pathlength),
                                 color = factor(pathlength)), se = FALSE)+
  theme_few()+
  #facet_wrap(~subject)+
  xlab("Raw RT") + ylab("Count") +
  ggtitle("Raw RT Histogram for All Trials")
```

## Raw RT Histogram for All Trials



## Linear Models

```
library(lme4)

## Loading required package: Matrix
new_netdemask_z$pathlengthfac = ordered(as.factor(as.character(new_netdemask_z$pathlength)),


RTprime_model = lmer(data = new_netdemask_z,
                        zRTTarget_trim ~ zRTPrime_trim +
                          (1|subject) + (1|ItemNumber))
summary(RTprime_model)

## Linear mixed model fit by REML ['lmerMod']
## Formula: zRTTarget_trim ~ zRTPrime_trim + (1 | subject) + (1 | ItemNumber)
##    Data: new_netdemask_z
##
## REML criterion at convergence: 2598.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.8426 -0.5540 -0.2503  0.4145  3.6542
##
## Random effects:
##  Groups     Name         Variance Std.Dev.
```

```
##  ItemNumber (Intercept) 0.2345    0.4842
##  subject    (Intercept) 0.0000    0.0000
##  Residual               0.7347    0.8571
## Number of obs: 928, groups:  ItemNumber, 699; subject, 4
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept)   -0.007133   0.033992   -0.21
## zRTPrime_trim  0.134093   0.032709    4.10
##
## Correlation of Fixed Effects:
##            (Intr)
## zRTPrim_trm 0.004
```

```r
contrasts(new_netdemask_z$pathlengthfac) = contr.treatment(6, base = 3)
RTprime_model_2 = lmer(data = new_netdemask_z,
                       zRTTarget_trim ~ zRTPrime_trim*pathlengthfac +
                         (1|subject) + (1|ItemNumber))
summary(RTprime_model_2)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: zRTTarget_trim ~ zRTPrime_trim * pathlengthfac + (1 | subject) +
##     (1 | ItemNumber)
##    Data: new_netdemask_z
##
## REML criterion at convergence: 2606.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.1039 -0.5650 -0.2221  0.3942  3.4911
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  ItemNumber (Intercept) 0.2381   0.4880
##  subject    (Intercept) 0.0000   0.0000
##  Residual               0.7207   0.8489
## Number of obs: 928, groups:  ItemNumber, 699; subject, 4
##
## Fixed effects:
##                            Estimate Std. Error t value
## (Intercept)                 0.11426    0.08404   1.360
## zRTPrime_trim               0.29683    0.08280   3.585
## pathlengthfac1             -0.31017    0.11795  -2.630
## pathlengthfac2             -0.07938    0.11768  -0.675
## pathlengthfac4             -0.16419    0.11807  -1.391
## pathlengthfac5              0.02148    0.11820   0.182
## pathlengthfac6             -0.16632    0.11822  -1.407
## zRTPrime_trim:pathlengthfac1 -0.22655    0.11328  -2.000
## zRTPrime_trim:pathlengthfac2 -0.20368    0.11604  -1.755
## zRTPrime_trim:pathlengthfac4 -0.28436    0.11781  -2.414
## zRTPrime_trim:pathlengthfac5 -0.03576    0.11243  -0.318
## zRTPrime_trim:pathlengthfac6 -0.22572    0.11407  -1.979
##
## Correlation of Fixed Effects:
##            (Intr) zRTPr_ pthln1 pthln2 pthln4 pthln5 pthln6 zRTP_:1
```

```
## zRTPrim_trm  0.082
## pthlngthfc1 -0.713 -0.058
## pthlngthfc2 -0.714 -0.059  0.509
## pthlngthfc4 -0.712 -0.058  0.507  0.508
## pthlngthfc5 -0.711 -0.058  0.507  0.508  0.506
## pthlngthfc6 -0.711 -0.058  0.507  0.508  0.506  0.505
## zRTPrm_tr:1 -0.060 -0.731  0.055  0.043  0.043  0.043  0.043
## zRTPrm_tr:2 -0.059 -0.714  0.042  0.011  0.042  0.042  0.042  0.522
## zRTPrm_tr:4 -0.058 -0.703  0.041  0.041 -0.003  0.041  0.041  0.514
## zRTPrm_tr:5 -0.060 -0.736  0.043  0.043  0.043  0.057  0.043  0.538
## zRTPrm_tr:6 -0.060 -0.726  0.042  0.043  0.042  0.042  0.060  0.531
##            zRTP_:2 zRTP_:4 zRTP_:5
## zRTPrim_trm
## pthlngthfc1
## pthlngthfc2
## pthlngthfc4
## pthlngthfc5
## pthlngthfc6
## zRTPrm_tr:1
## zRTPrm_tr:2
## zRTPrm_tr:4  0.502
## zRTPrm_tr:5  0.526   0.518
## zRTPrm_tr:6  0.518   0.510   0.535
```

```r
car::Anova(RTprime_model_2)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: zRTTarget_trim
##                          Chisq Df Pr(>Chisq)
## zRTPrime_trim            17.281  1  3.223e-05 ***
## pathlengthfac            10.471  5    0.06294 .
## zRTPrime_trim:pathlengthfac 10.334  5    0.06631 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
options(contrasts = c('contr.sum', 'contr.poly'))
 library(lsmeans)
 library(multcomp)
 demasking_lsm = lsmeans::lsmeans(RTprime_model_2, c("zRTPrime_trim",
                                                     "pathlengthfac" ))
```

```
## Loading required namespace: lmerTest
```

```r
 prime_effect = cld(demasking_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 |
|----|----------|----------|----|----|---------|---------|
| 7  | -0.00644373153009005,3 - -0.00644373153009005,1 | 0.3087111 | 0.1179091 | 640.4404 | 2.618214 | 0.0943294 |
| 11 | -0.00644373153009005,6 - -0.00644373153009005,1 | 0.3304238 | 0.1172789 | 632.7127 | 2.817419 | 0.0560098 |

```
RTprime_model_3 = lmer(data = new_netdemask_z,
                       zRTTarget_trim ~ pathlengthfac +
                       (1|subject) + (1|ItemNumber))
summary(RTprime_model_3)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: zRTTarget_trim ~ pathlengthfac + (1 | subject) + (1 | ItemNumber)
##    Data: new_netdemask_z
##
## REML criterion at convergence: 2614.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.9212 -0.5755 -0.2288  0.4005  3.5641
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  ItemNumber (Intercept) 0.2347   0.4845
##  subject    (Intercept) 0.0000   0.0000
##  Residual               0.7454   0.8634
## Number of obs: 928, groups:  ItemNumber, 699; subject, 4
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept)    0.08936    0.08455   1.057
## pathlengthfac1 -0.28718    0.11884  -2.417
## pathlengthfac2 -0.04856    0.11847  -0.410
## pathlengthfac4 -0.13836    0.11876  -1.165
## pathlengthfac5  0.03796    0.11910   0.319
## pathlengthfac6 -0.14387    0.11910  -1.208
##
## Correlation of Fixed Effects:
##            (Intr) pthln1 pthln2 pthln4 pthln5
## pthlngthfc1 -0.711
## pthlngthfc2 -0.714  0.508
## pthlngthfc4 -0.712  0.507  0.508
## pthlngthfc5 -0.710  0.505  0.507  0.505
## pthlngthfc6 -0.710  0.505  0.507  0.505  0.504
```

```
car::Anova(RTprime_model_3)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: zRTTarget_trim
##               Chisq Df Pr(>Chisq)
## pathlengthfac 9.9904  5    0.07551 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(RTprime_model_3, RTprime_model_2)
```

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

```
## Data: new_netdemask_z
## Models:
## RTprime_model_3: zRTTarget_trim ~ pathlengthfac + (1 | subject) + (1 | ItemNumber)
```

```
## RTprime_model_2: zRTTarget_trim ~ zRTPrime_trim * pathlengthfac + (1 | subject) +
## RTprime_model_2:     (1 | ItemNumber)
##                Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## RTprime_model_3  9 2613.3 2656.8 -1297.7   2595.3
## RTprime_model_2 15 2597.8 2670.3 -1283.9   2567.8 27.519      6  0.0001157
##
## RTprime_model_3
## RTprime_model_2 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## Plot

```
primeplot = new_netdemask_z %>%
  ggplot(aes(x = zRTPrime_trim, y = zRTTarget_trim,
             group = factor(subject))) +
  geom_smooth(method = "lm", se = FALSE, color = "forestgreen", size = 0.5)+
    xlab("z-RT to Demask Prime") + ylab ("z-RT to Demask Target")+
  ggtitle("Target Retrieval Accuracy by \nPrime Demasking RT & Prime Condition")+
theme_hc() +
  facet_wrap(~pathlengthfac)+
    theme(axis.text = element_text(face = "bold", size = rel(1.2)),
          axis.title = element_text(face = "bold", size = rel(1.2)),
          legend.title = element_text(face = "bold", size = rel(1.2)),
                strip.text.x = element_text(face = "bold", size = rel(1.4)),
          plot.title = element_text(face = "bold", size = rel(1.2), hjust = .5))

primeplot + stat_smooth(aes(group = pathlengthfac, color = "red"), method = "lm", se = FALSE)
```

**Target Retrieval Accuracy by**
**Prime Demasking RT & Prime Condition**
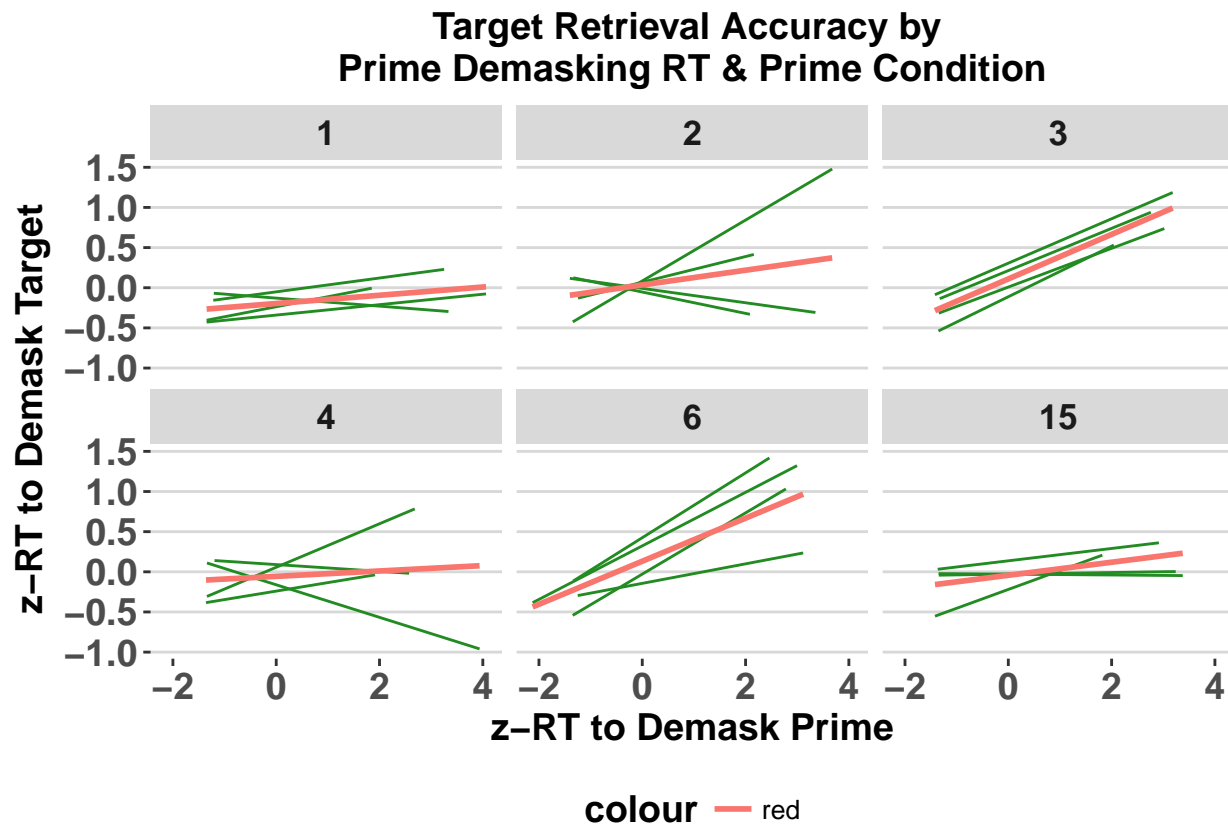


```
## Main effects
new_netdemask_z %>%
  ggplot(aes(x = zRTPrime_trim, y = zRTTarget_trim))+
#  geom_smooth(method = "loess")+
  geom_smooth(size = 1, color = "blue", method = "lm")+
 theme_few()+
  #scale_x_continuous(breaks = c(1,2,3,4,5,6,10,15,20))+
    xlab("RT to recognize Prime") + ylab("RT to Recognise Target") +
  ggtitle("Pure Demasking RT")  +
  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)))
```

## Pure Demasking RT
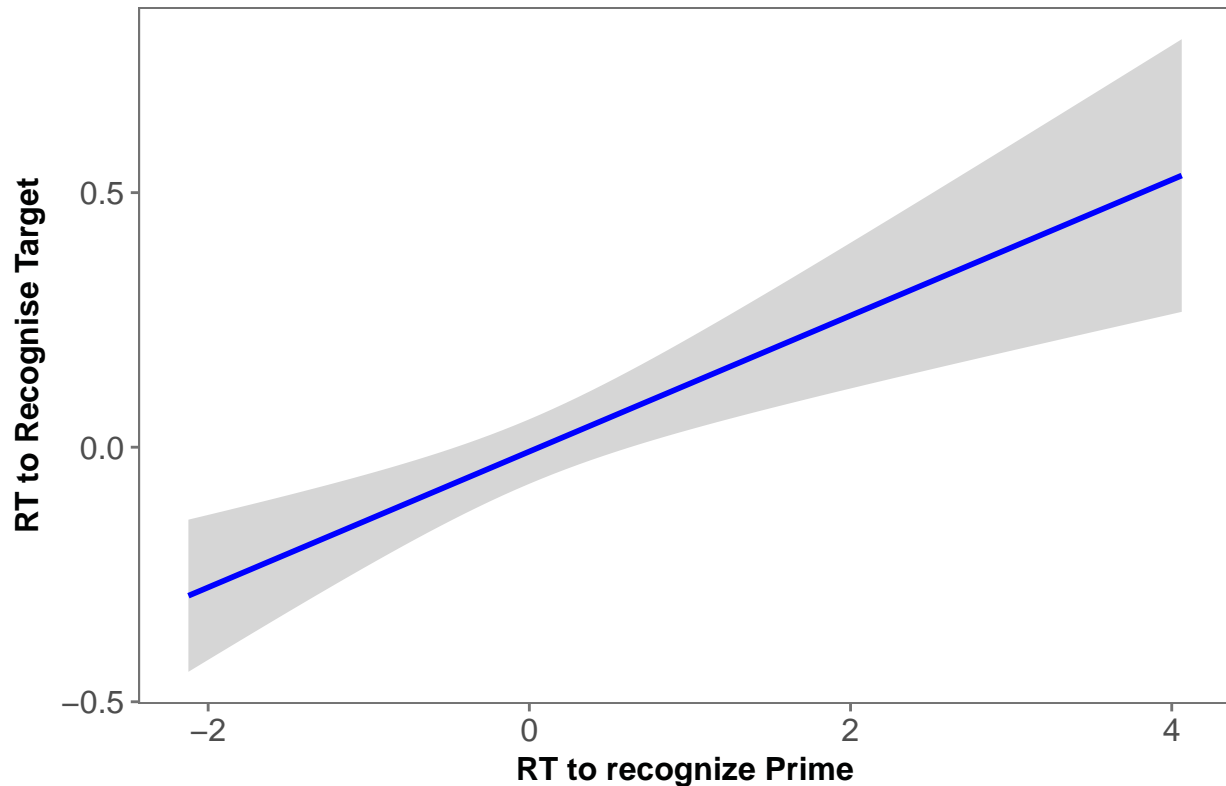


```r
path_group = Rmisc::summarySE(new_netdemask_z,
                        measurevar = "zRTTarget_trim",
                        groupvars = c("pathlengthfac"))

path_group %>%
  ggplot(aes(x = pathlengthfac, y = zRTTarget_trim))+
# geom_smooth(method = "loess")+
  geom_bar(stat = "identity", position = "dodge", width = 0.5,
          color = "black", fill = "darkslategray4")+
  geom_errorbar(aes(ymin=zRTTarget_trim - ci,
                    ymax=zRTTarget_trim + 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("z-RT to recognise Target") +
  ggtitle("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)))
```

RT to Demask Target by Path Length