

# Network Demasking

## Reading the Data

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

## Raw Reaction Time

```
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

```
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 11 8788695  798972
##
## Error: subject:pathlengthfac
##           Df Sum Sq Mean Sq F value Pr(>F)
## pathlengthfac  5 207507   41501   4.261 0.0024 **
## Residuals     55 535695    9740
## ---
## 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)),

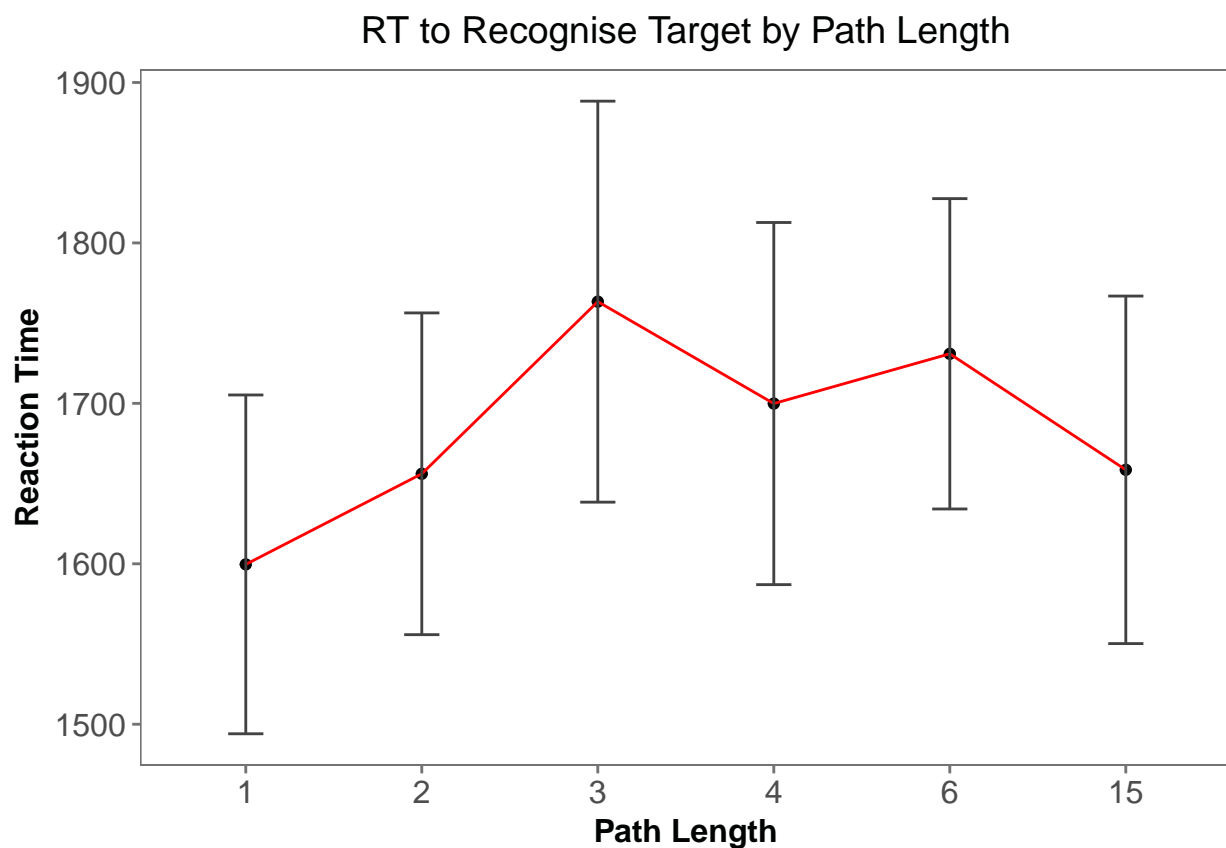
library(ggplot2)
library(ggthemes)

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

```

library(ggplot2)
library(ggthemes)

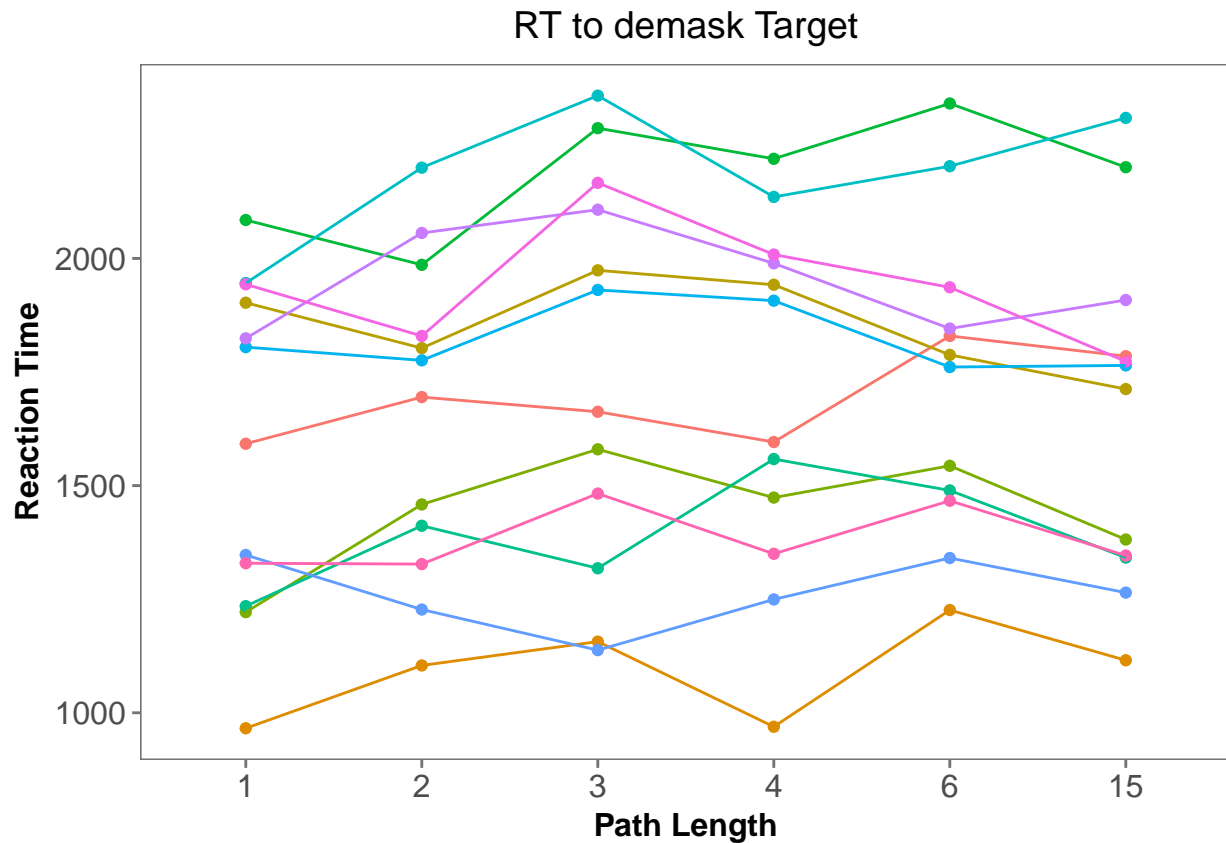
netdemask_rt %>%
  ggplot(aes(x = pathlengthfac, y = RTRecogniseTarget,
             group = subject, color = subject))+
  geom_point()+
  geom_line()+

```

```

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

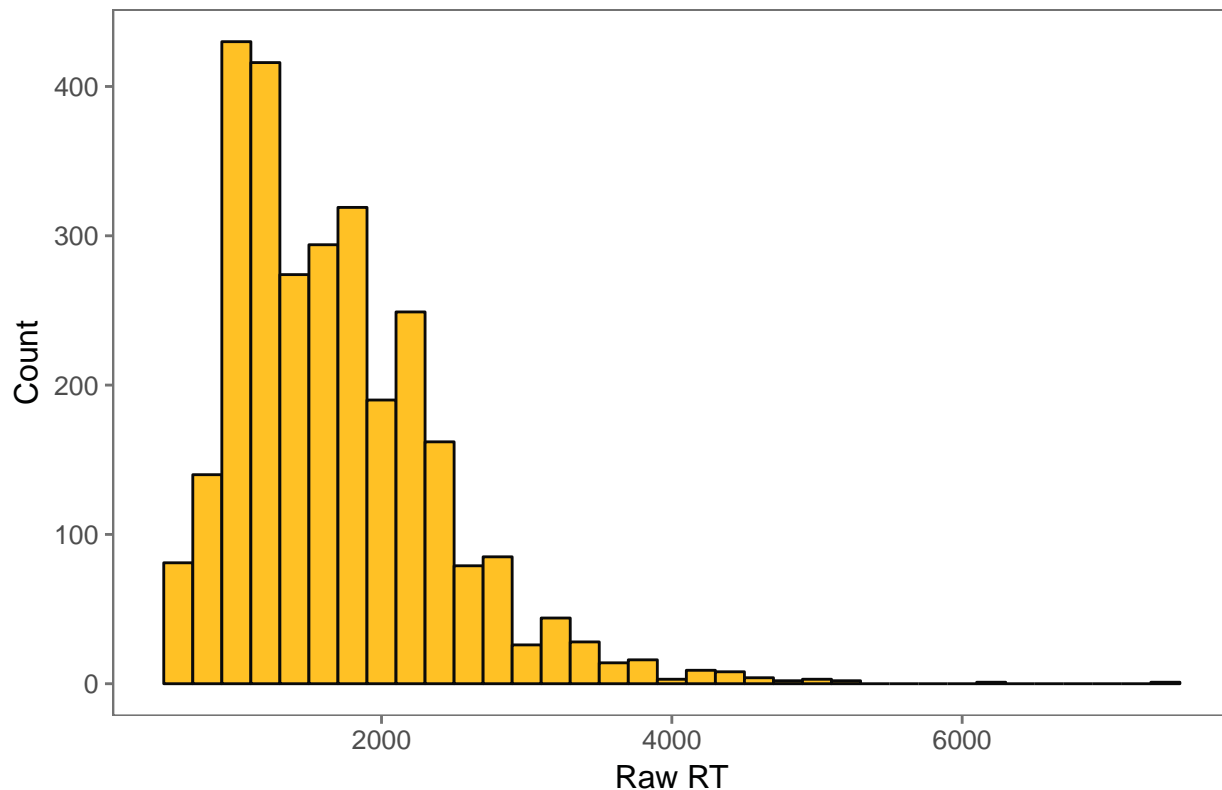
### Histogram of RT

```

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)

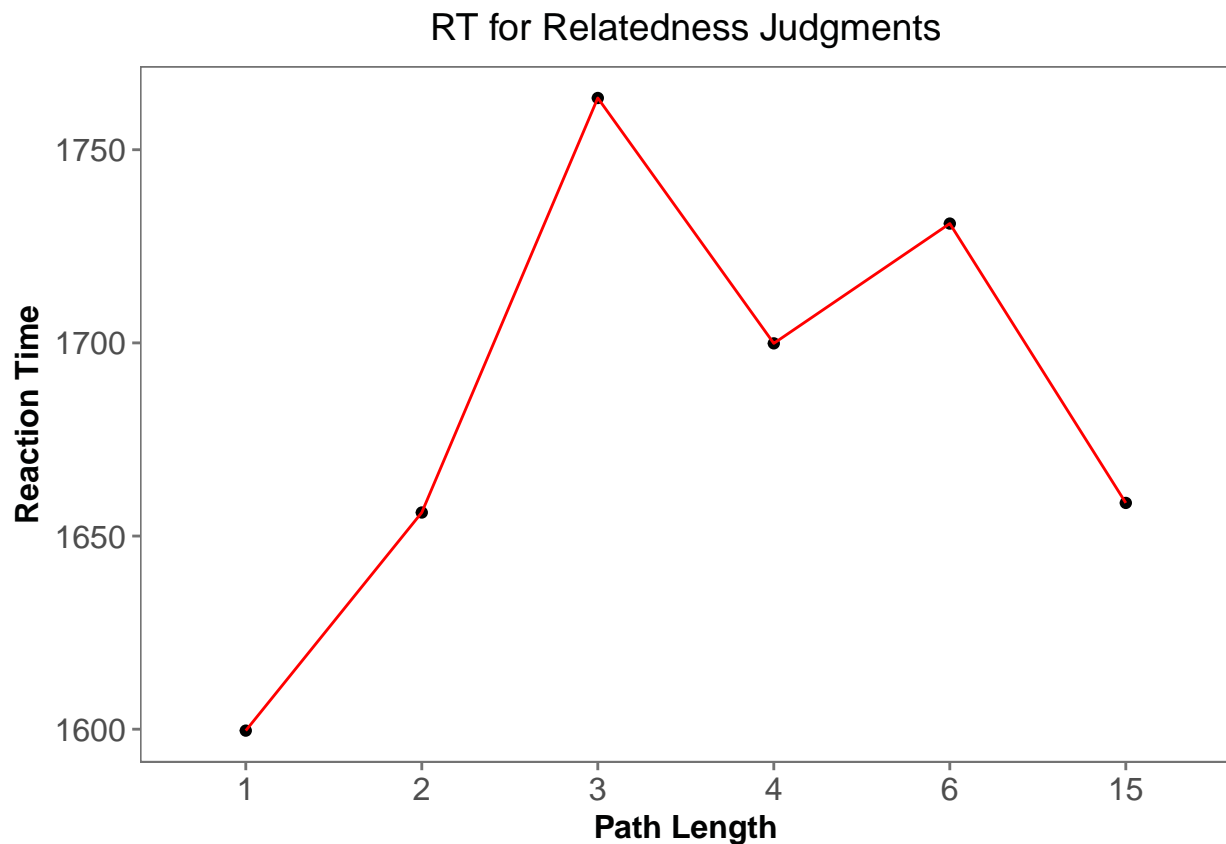
netdemask_rt_agg_firsttrim = group_by(netdemask_firsttrim, pathlength ) %>%
  summarise_at(vars(RTRecognisePrime, RTRecogniseTarget), mean)

netdemask_rt_agg_firsttrim$pathlengthfac = ordered(as.factor(as.character(netdemask_rt_agg_firsttrim$pathlength)))

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)

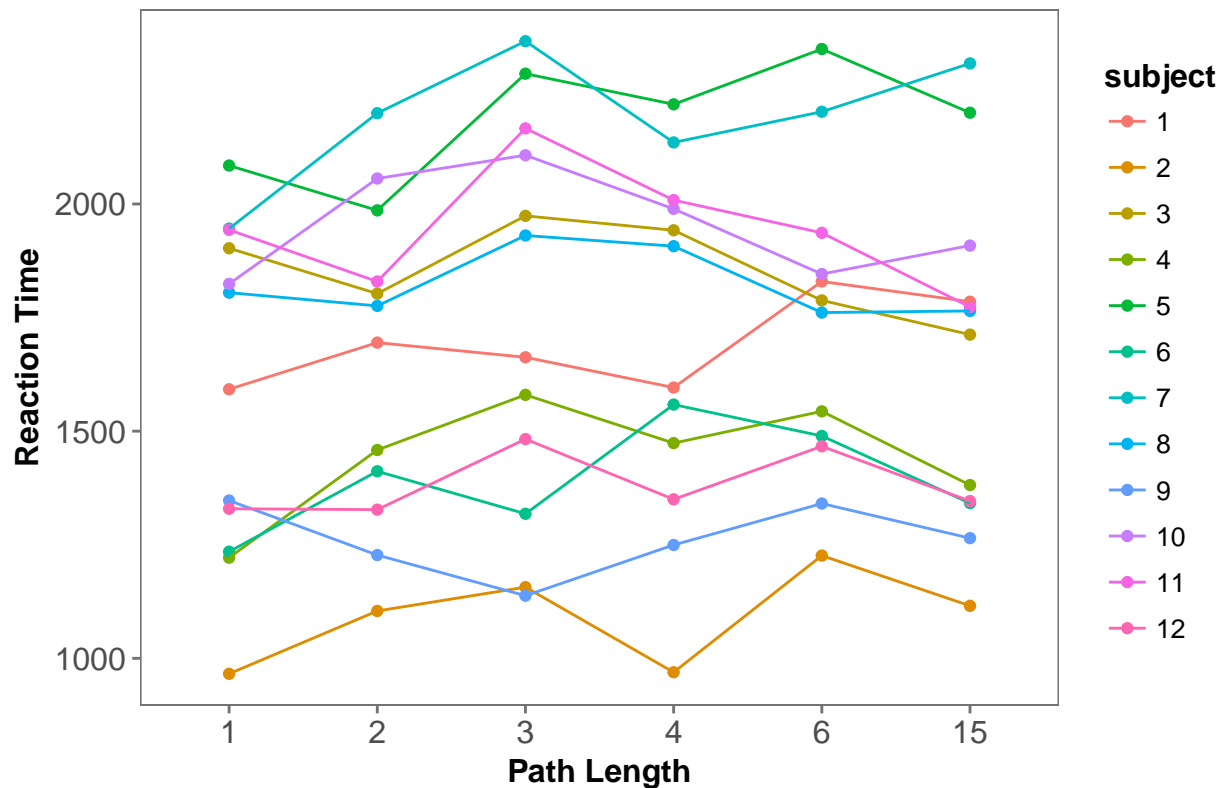
netdemask_rt_firsttrim$pathlengthfac = ordered(as.factor(as.character(netdemask_rt_firsttrim$pathlength.
  levels = c("1", "2", "3", "4", "6", "15")))
netdemask_rt_firsttrim$subject = as.factor(netdemask_rt_firsttrim$subject)
netdemask_rt_firsttrim %>%
  ggplot(aes(x = pathlengthfac, y = RTRecogniseTarget,
    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)))

```

## RT for Relatedness Judgments



## 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")

```

```

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

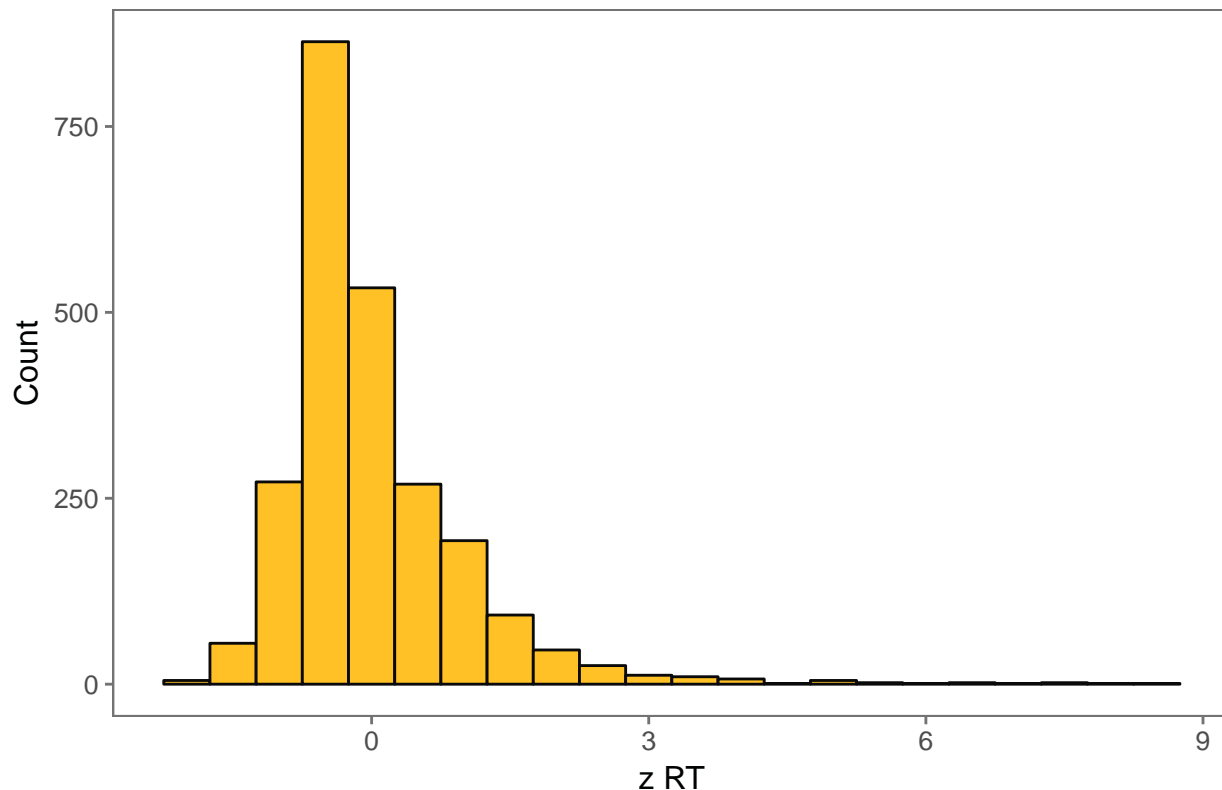
```

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")

```

```
## Warning: Removed 480 rows containing non-finite values (stat_bin).
```

## 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
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

```
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

```
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 0.006821 0.0007579
##
## Error: subject:pathlengthfac
##           Df Sum Sq Mean Sq F value  Pr(>F)
## pathlengthfac  5 0.5839 0.11678   3.581 0.00823 **
## Residuals     45 1.4674 0.03261
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

```
## The 'lsmmeans' package is being deprecated.
## Users are encouraged to switch to 'emmeans'.
## See help('transition') for more information, including how
## to convert 'lsmmeans' 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
7	3 - 1	0.2756737	0.0807565	45	3.413639	0.0161535
11	6 - 1	0.2975902	0.0807565	45	3.685029	0.0075769

## Plotting RTs: collapsed

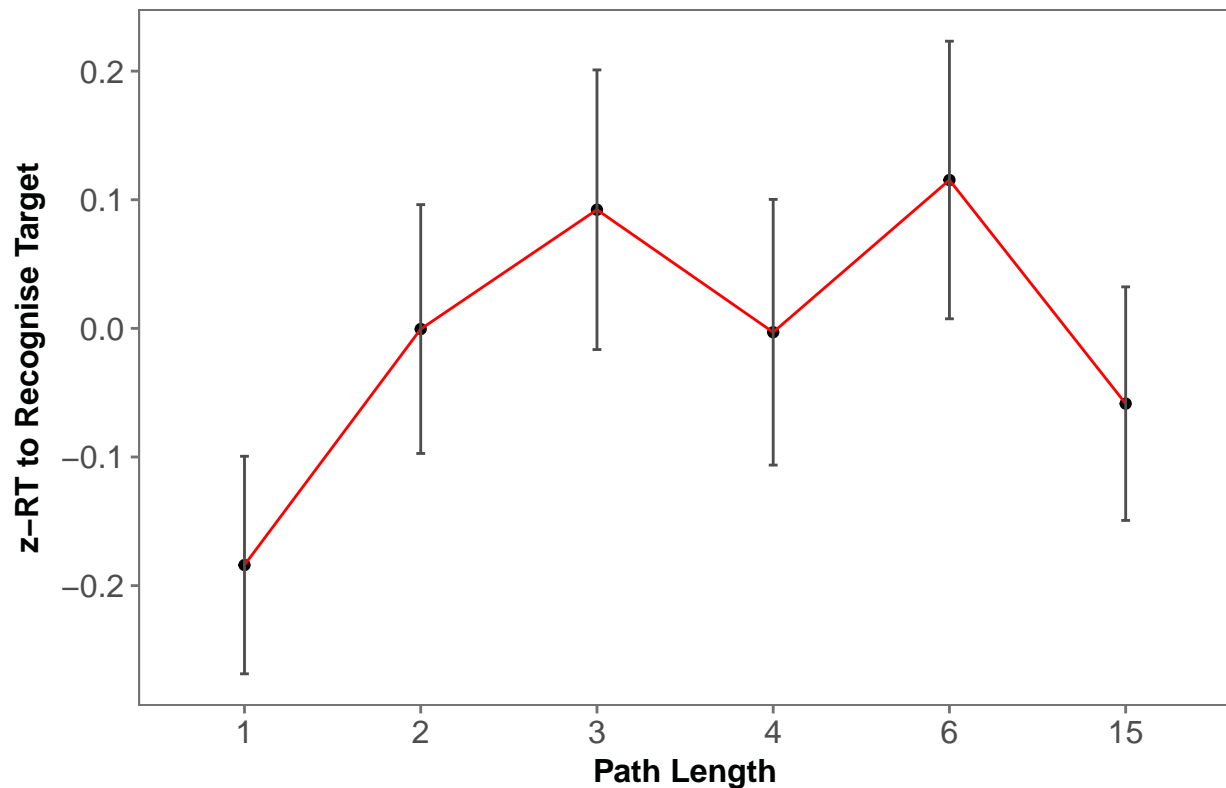
```
z_rmisc$pathlengthfac = ordered(as.factor(as.character(z_rmisc$pathlength)),

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)))
```

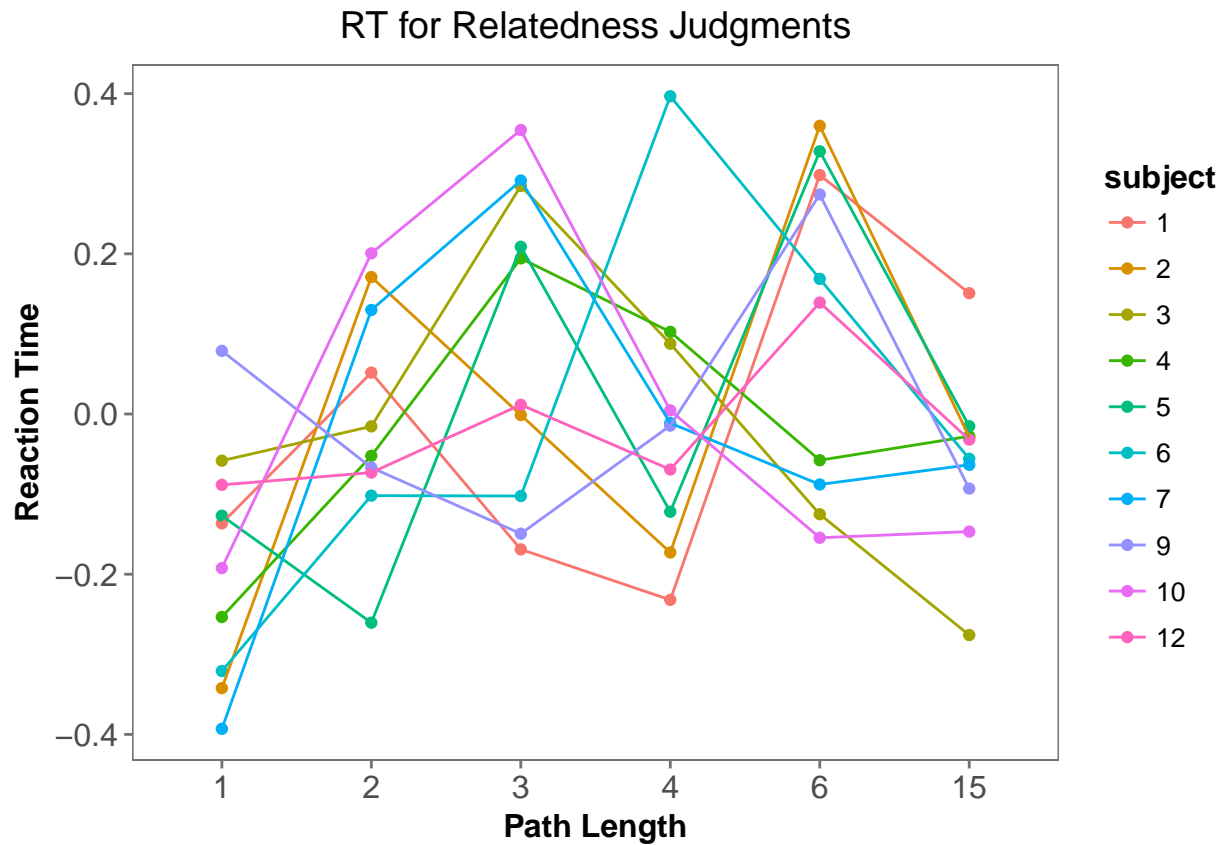
z-RT to Demask Target by Path Length



## 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

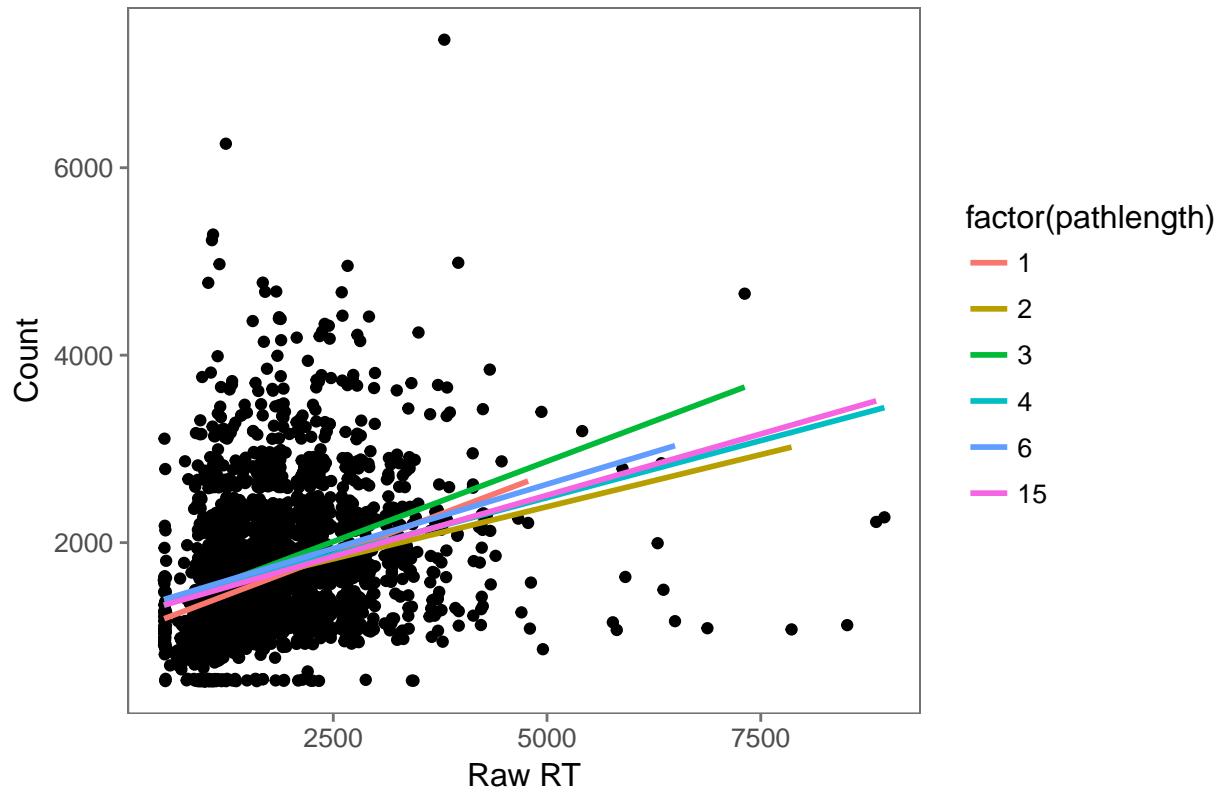
### Simple Scatter Plot

```
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")
```

## Warning: Removed 2 rows containing non-finite values (stat\_smooth).

## Warning: Removed 2 rows containing missing values (geom\_point).

## 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: 6373.3
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.5237 -0.6009 -0.1671  0.4519  3.8842
##
## Random effects:
## Groups      Name                Variance Std.Dev.
```

```

## ItemNumber (Intercept) 0.2175 0.4664
## subject (Intercept) 0.0000 0.0000
## Residual 0.7419 0.8613
## Number of obs: 2321, groups: ItemNumber, 718; subject, 10
##
## Fixed effects:
## Estimate Std. Error t value
## (Intercept) -0.002003 0.025289 -0.079
## zRTPrime_trim 0.193516 0.020238 9.562
##
## Correlation of Fixed Effects:
## (Intr)
## zRTPrim_trm 0.001

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: 6390.4
##
## Scaled residuals:
## Min 1Q Median 3Q Max
## -2.4047 -0.6037 -0.1668 0.4474 3.7863
##
## Random effects:
## Groups Name Variance Std.Dev.
## ItemNumber (Intercept) 0.2115 0.4599
## subject (Intercept) 0.0000 0.0000
## Residual 0.7421 0.8614
## Number of obs: 2321, groups: ItemNumber, 718; subject, 10
##
## Fixed effects:
## Estimate Std. Error t value
## (Intercept) 0.10305 0.06197 1.663
## zRTPrime_trim 0.23403 0.04790 4.886
## pathlengthfac1 -0.27935 0.08723 -3.202
## pathlengthfac2 -0.10650 0.08708 -1.223
## pathlengthfac4 -0.11485 0.08732 -1.315
## pathlengthfac5 0.03192 0.08732 0.366
## pathlengthfac6 -0.16039 0.08743 -1.835
## zRTPrime_trim:pathlengthfac1 -0.07130 0.06933 -1.028
## zRTPrime_trim:pathlengthfac2 -0.07514 0.06967 -1.079
## zRTPrime_trim:pathlengthfac4 -0.02668 0.07054 -0.378
## zRTPrime_trim:pathlengthfac5 -0.01007 0.06925 -0.145
## zRTPrime_trim:pathlengthfac6 -0.06586 0.06715 -0.981
##
## Correlation of Fixed Effects:
## (Intr) zRTPr_ pthln1 pthln2 pthln4 pthln5 pthln6 zRTP_:1

```

```
## zRTPrim_trm 0.020
## pthlngthfc1 -0.710 -0.014
## pthlngthfc2 -0.712 -0.014 0.505
## pthlngthfc4 -0.710 -0.014 0.504 0.505
## pthlngthfc5 -0.710 -0.014 0.504 0.505 0.504
## pthlngthfc6 -0.709 -0.014 0.503 0.504 0.503 0.503
## zRTPrm_tr:1 -0.014 -0.691 0.027 0.010 0.010 0.010 0.010
## zRTPrm_tr:2 -0.014 -0.688 0.010 0.009 0.010 0.010 0.010 0.475
## zRTPrm_tr:4 -0.013 -0.679 0.009 0.009 0.002 0.009 0.009 0.469
## zRTPrm_tr:5 -0.014 -0.692 0.010 0.010 0.010 0.001 0.010 0.478
## zRTPrm_tr:6 -0.014 -0.713 0.010 0.010 0.010 0.010 0.003 0.493
##          zRTP_:2 zRTP_:4 zRTP_:5
## zRTPrim_trm
## pthlngthfc1
## pthlngthfc2
## pthlngthfc4
## pthlngthfc5
## pthlngthfc6
## zRTPrm_tr:1
## zRTPrm_tr:2
## zRTPrm_tr:4 0.467
## zRTPrm_tr:5 0.476 0.470
## zRTPrm_tr:6 0.491 0.484 0.493
```

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

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: zRTTarget_trim
##              Chisq Df Pr(>Chisq)
## zRTPrime_trim      90.8469 1 < 2.2e-16 ***
## pathlengthfac      16.6158 5 0.005289 **
## zRTPrime_trim:pathlengthfac 2.2912 5 0.807558
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

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

```
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	-0.00516301671575697,3 - -0.00516301671575697,1	0.2789798	0.0872208	673.9467	3.198546	0.0180460
11	-0.00516301671575697,6 - -0.00516301671575697,1	0.3109492	0.0869135	672.3820	3.577686	0.0049764



```

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: 6456.6
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.5057 -0.6111 -0.1837  0.4541  3.8165
##
## Random effects:
## Groups      Name                Variance Std.Dev.
## ItemNumber (Intercept) 0.2058    0.4536
## subject     (Intercept) 0.0000    0.0000
## Residual                0.7781    0.8821
## Number of obs: 2321, groups: ItemNumber, 718; subject, 10
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)    0.09681    0.06233   1.553
## pathlengthfac1 -0.28009    0.08771  -3.193
## pathlengthfac2 -0.09982    0.08759  -1.140
## pathlengthfac4 -0.10470    0.08782  -1.192
## pathlengthfac5  0.04158    0.08783   0.473
## pathlengthfac6 -0.15103    0.08794  -1.717
##
## Correlation of Fixed Effects:
##              (Intr) pthln1 pthln2 pthln4 pthln5
## pthlngthfc1 -0.711
## pthlngthfc2 -0.712  0.506
## pthlngthfc4 -0.710  0.504  0.505
## pthlngthfc5 -0.710  0.504  0.505  0.504
## pthlngthfc6 -0.709  0.504  0.504  0.503  0.503
car::Anova(RTprime_model_3)

## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: zRTTarget_trim
##              Chisq Df Pr(>Chisq)
## pathlengthfac 17.005  5    0.00449 **
## ---
## 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)

```

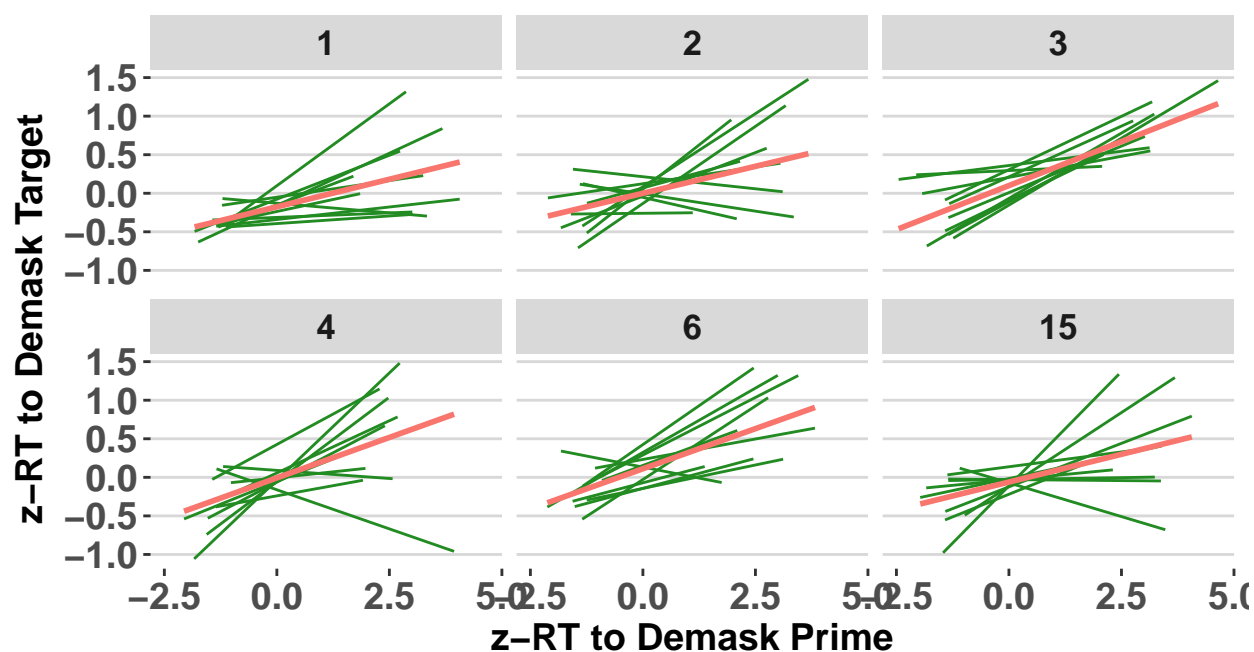
```
## RTprime_model_2: zRTTarget_trim ~ zRTPrime_trim * pathlengthfac + (1 | subject) +
## RTprime_model_2:      (1 | ItemNumber)
##           Df      AIC      BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## RTprime_model_3   9 6452.2 6504.0 -3217.1   6434.2
## RTprime_model_2  15 6372.9 6459.1 -3171.4   6342.9 91.337      6 < 2.2e-16
##
## 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

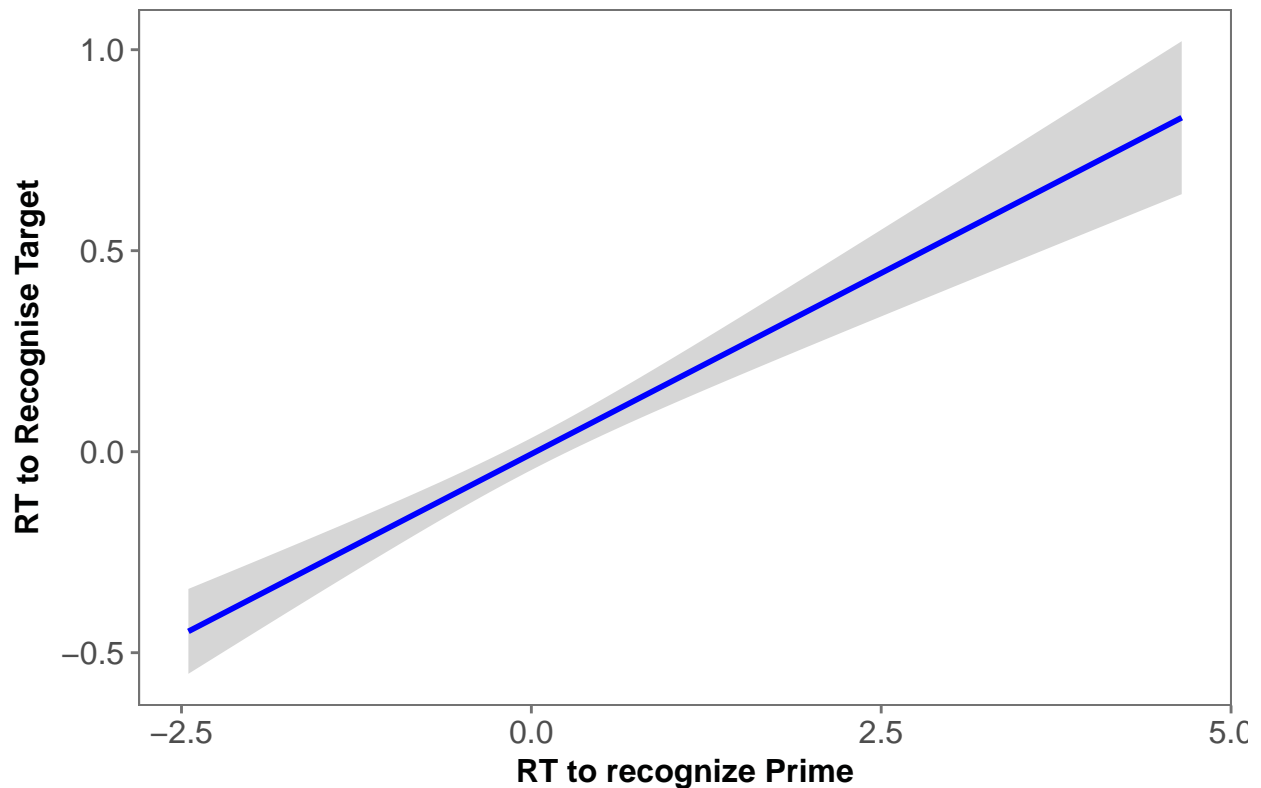


colour — red

## 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



```
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)))
```

