ConfirmCLT

Load libraries

```
library("ggplot2") # general plotting
library("ggpubr") # combining plots
```

Specify the experiment parameters

```
ntimes <- 5000 # number of noise replicates to investigate normality
maxdim <- 10000 # maximal dimension of the data set to be investigated
dims <- round(exp(seq(log(2), log(maxdim), length.out=10))) # dimensions to study from log-scale
a <- 0.75 # magnitude of noise: per dimension we sample noise uniformly from [-a, a]
```

Specify the ground truth sequences

```
sequences <- list()

# bounded 2-norm, bounded infinity-norm
sequences[[1]] <- rbind(rep(0, maxdim), rep(0, maxdim), c(1, rep(0, maxdim - 1)))

# increasing 2-norm, bounded infinity-norm
sequences[[2]] <- rbind(rep(0, maxdim), rep(0, maxdim), rep(1, maxdim))

# increasing 2-norm, increasing infinity-norm
sequences[[3]] <- rbind(rep(0, maxdim), rep(0, maxdim), (1:maxdim)^(1/4 - 0.01))
```

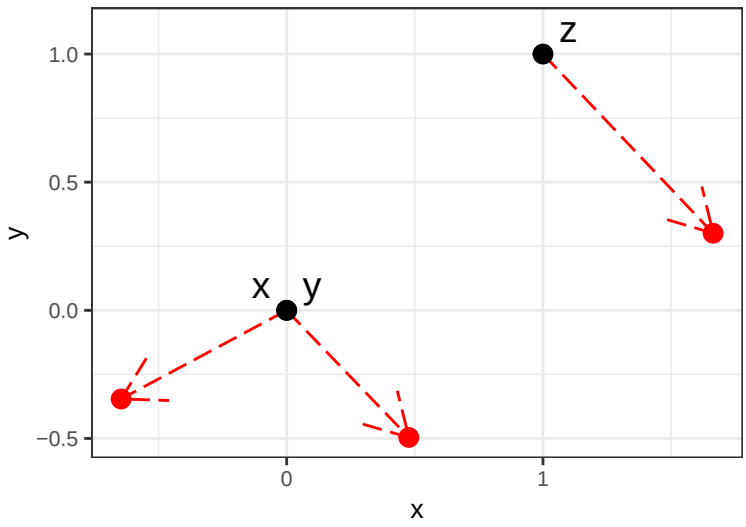
Choose an example triple of sequences for illustration

```
idx <- 2
x <- sequences[[idx]][1,]; y <- sequences[[idx]][2,]; z <- sequences[[idx]][3,]
```

We now view the magnitude of noise we will use for the initial two dimensions

```
set.seed(12)
X <- data.frame(x=c(x[1], y[1], z[1]), y=c(x[2], y[2], z[2]))
XN <- X + matrix(runif(2 * 3, min=-a, max=a), ncol=2)

ggplot(X, aes(x=x, y=y)) +
  geom_segment(x=X[,1], y=X[,2], xend=XN[,1], yend=XN[,2], color="red", arrow=arrow(), linetype=5) +
  geom_point(size=3) +
  geom_point(data=XN, col="red", size=3) +
  theme_bw() +
  theme(text=element_text(size=10)) +
  coord_fixed() +
  annotate("text", x=X[,1] + c(-.1, .1, .1), y=X[,2] + c(.1, .1, .1), label=c("x", "y", "z"), size=5)
```



Calculate the variance and fourth moment of our random noise variable

```
sigma2 <- a^2 / 3 # variance of uniform distribution over [-a, a]
mu4 <- a^4 / 5 # fourth moment of uniform distribution over [-a, a]
```

Investigate normality of Y^(d) with added two-dimensional noise

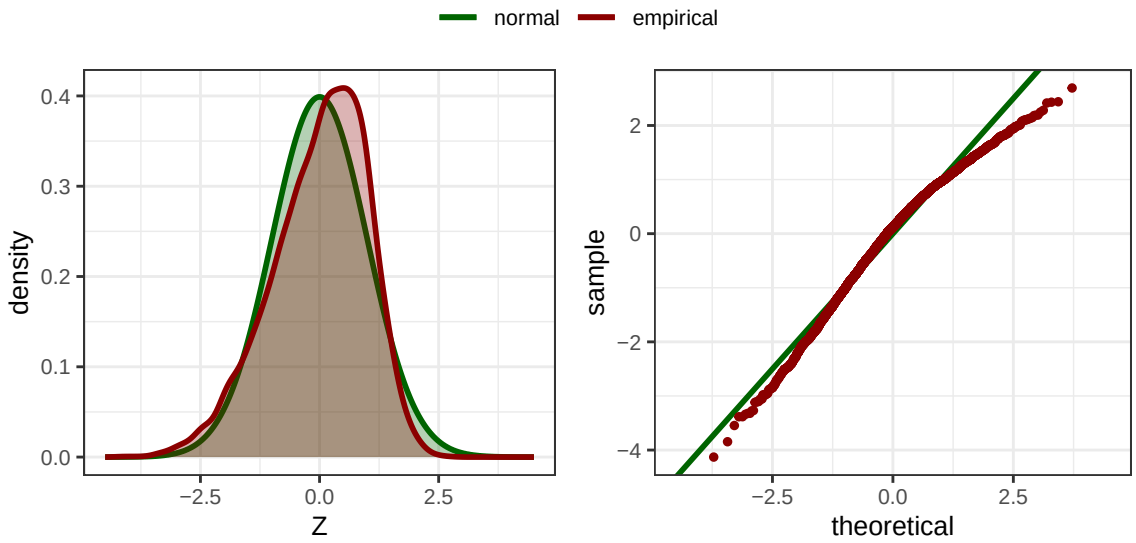
```
this_dim <- 2
xd <- x[1:this_dim]; yd <- y[1:this_dim]; zd <- z[1:this_dim]
dxy <- norm(xd - yd, type="2"); dxz <- norm(xd - zd, type="2")
mu_d <- dxy^2 - dxz^2
sigma2_d <- 2 * this_dim * (mu4 + 3 * sigma2^2) +
  8 * sigma2*(dxy^2 + dxz^2 - sum((xd - yd) * (xd - zd)))

Z <- sapply(1:ntimes, function(i){
    N <- matrix(runif(this_dim * 3, min=-a, max=a), ncol=3)
    dnxy <- norm(xd + N[,1] - yd - N[,2], type="2"); dnxz <- norm(xd + N[,1] - zd - N[,3], type="2")
    return((dnxy^2 - dnxz^2 - mu_d) / sqrt(sigma2_d))
  })

P1 <- ggplot(data.frame(Z=Z), aes(x=Z)) +
  stat_function(fun=dnorm, n=ntimes, geom="area", alpha=0.3, fill="darkgreen") +
  stat_function(fun=dnorm, n=ntimes, geom="line", aes(col="1"), size=1) +
  geom_density(alpha=.3, fill="darkred", col=NA) +
  geom_line(stat="density", aes(col="2"), size=1) +
  xlim(c(-4.5, 4.5)) +
  ylab("density") +
  scale_colour_manual(name="", labels=c("normal", "empirical"), values=c("darkgreen", "darkred")) +
  theme_bw() +
  theme(text=element_text(size=10))

P2 <- ggplot(data.frame(Z=Z), aes(sample=Z)) +
  geom_abline(aes(intercept=0, slope=1, col="1"), size=1) +
  stat_qq(aes(col="2"), size=1) +
  scale_colour_manual(name="", labels=c("normal", "empirical"), values=c("darkgreen", "darkred")) +
  xlim(c(-4.5, 4.5)) +
  theme_bw() +
  theme(text=element_text(size=10))
```

```
ggarrange(P1, P2, ncol=2, common.legend=TRUE)
```



Investigate normality of Yˆ(d) with added high-dimensional noise

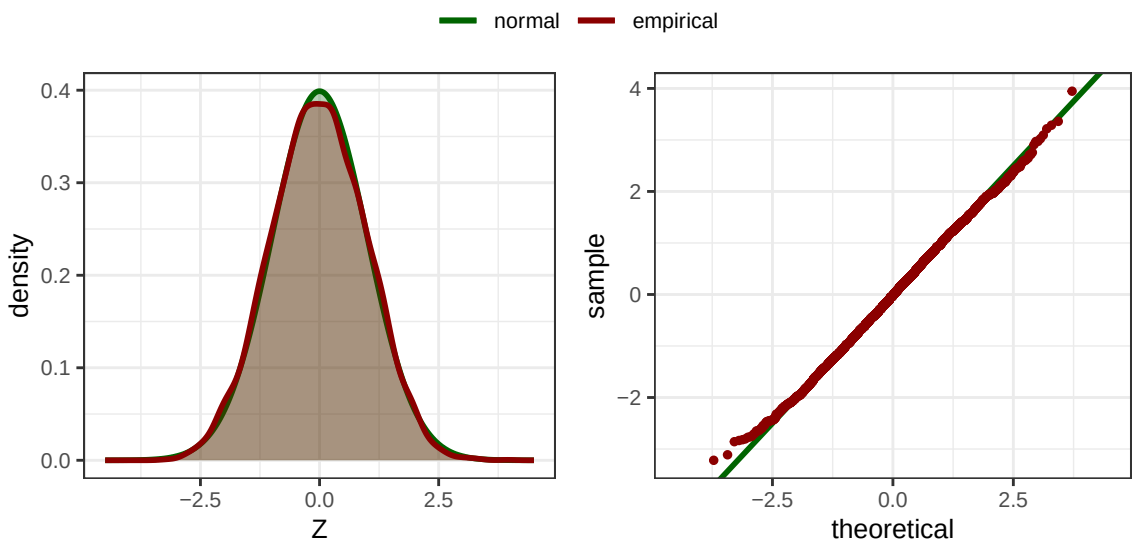
```
this_dim <- maxdim
xd <- x[1:this_dim]; yd <- y[1:this_dim]; zd <- z[1:this_dim]
dxy <- norm(xd - yd, type="2"); dxz <- norm(xd - zd, type="2")
mu_d <- dxy^2 - dxz^2
sigma2_d <- 2 * this_dim * (mu4 + 3 *sigma2^2) +
  8 * sigma2 * (dxy^2 + dxz^2 - sum((xd - yd) * (xd - zd)))

Z <- sapply(1:ntimes, function(i){
  N <- matrix(runif(this_dim * 3, min=-a, max=a), ncol=3)
  dnxy <- norm(xd + N[,1] - yd - N[,2], type="2"); dnxz <- norm(xd + N[,1] - zd - N[,3], type="2")
  return((dnxy^2 - dnxz^2 - mu_d) / sqrt(sigma2_d))
})

P1 <- ggplot(data.frame(Z=Z), aes(x=Z)) +
  stat_function(fun=dnorm, n=ntimes, geom="area", alpha=0.3, fill="darkgreen") +
  stat_function(fun=dnorm, n=ntimes, geom="line", aes(col="1"), size=1) +
  geom_density(alpha=.3, fill="darkred", col=NA) +
  geom_line(stat="density", aes(col="2"), size=1) +
  xlim(c(-4.5, 4.5)) +
  ylab("density") +
  scale_colour_manual(name="", labels=c("normal", "empirical"), values=c("darkgreen", "darkred")) +
  theme_bw() +
  theme(text=element_text(size=10))

P2 <- ggplot(data.frame(Z=Z), aes(sample=Z)) +
  geom_abline(aes(intercept=0, slope=1, col="1"), size=1) +
  stat_qq(aes(col="2"), size=1) +
  scale_colour_manual(name="", labels=c("normal", "empirical"), values=c("darkgreen", "darkred")) +
  xlim(c(-4.5, 4.5)) +
  theme_bw() +
  theme(text=element_text(size=10))

ggarrange(P1, P2, ncol=2, common.legend=TRUE)
```



Quantify the normality of Yˆ(d) for varying dimensions and types of sequences

```
shapiro_tests <- data.frame(sequence=integer(), dim=integer(), ST=numeric())
sequence_params <- lapply(1:length(sequences), function(idx){
  lapply(dims, function(this_dim){
    xd <- x[1:this_dim]; yd <- y[1:this_dim]; zd <- z[1:this_dim]
    dxy <- norm(xd - yd, type="2"); dxz <- norm(xd - zd, type="2")
    mu_d <- dxy^2 - dxz^2
    sigma2_d <- 2 * this_dim * (mu4 + 3 * sigma2^2) +
      8 * sigma2 * (dxy^2 + dxz^2 - sum((xd - yd) * (xd - zd)))
    return(list(mu_d=mu_d, sigma2_d=sigma2_d))
  })
})

for(idx1 in 1:length(sequences)){
  x <- sequences[[idx1]][1,]; y <- sequences[[idx1]][2,]; z <- sequences[[idx1]][3,]

  Zs <- sapply(1:ntimes, function(n){
    N <- matrix(runif(maxdim * 3, min=-a, max=a), ncol=3)
    sapply(1:length(dims), function(idx2){
      this_dim <- dims[idx2]
      dnxy <- norm(x[1:this_dim] + N[1:this_dim, 1] - y[1:this_dim] - N[1:this_dim, 2], type="2")
      dnxz <- norm(x[1:this_dim] + N[1:this_dim, 1] - z[1:this_dim] - N[1:this_dim, 3], type="2")
      return((dnxy^2 - dnxz^2 - sequence_params[[idx1]][[idx2]]$mu_d) /
               sqrt(sequence_params[[idx1]][[idx2]]$sigma2_d))
    })
  })

  ST <- sapply(apply(Zs, 1, shapiro.test), function(st) st[[1]])

  shapiro_tests[nrow(shapiro_tests) + 1:length(ST),] <- cbind(rep(idx1, length(ST)), dims, ST)
}

ggplot(shapiro_tests, aes(x=dim, y=ST, col=factor(sequence))) +
  geom_line() +
  geom_point() +
```

```
  scale_x_log10() +
  ylab("Shapiro-Wilk statistic") +
  scale_colour_manual(name="Convergence/Divergence",
                      labels=c(bquote(l[2]~bounded~--~l[infinity]~bounded),
                               bquote(l[2]~unbounded~--~l[infinity]~bounded),
                               bquote(l[2]~unbounded~--~l[infinity]~unbounded)),
                      values=c("#F8766D", "#7CAE00", "#00BFC4")) +
  theme_classic() +
  theme(text=element_text(size=10))
```