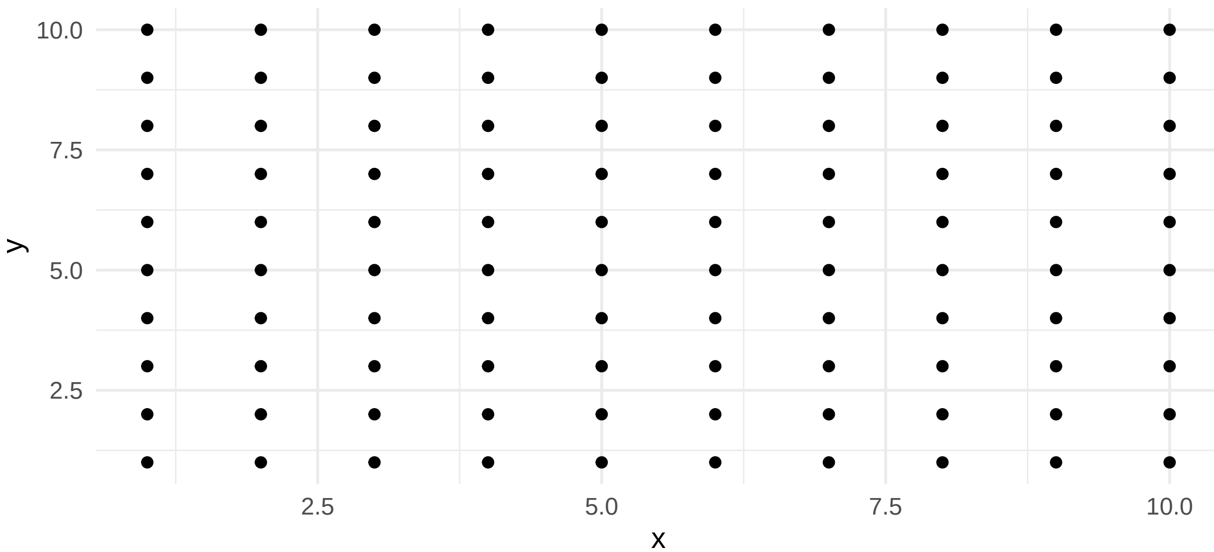
Sonish Lamsal / Bradley Thompson

STA6375: Computational Statistics I

Homework 3

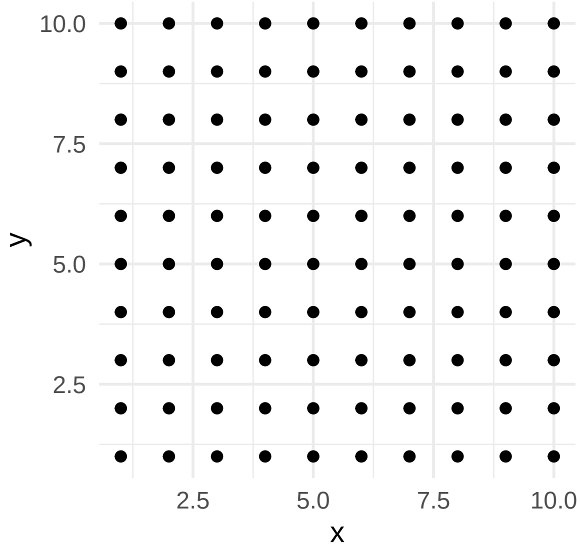
1a.

```
library("tidyverse")



df <- expand.grid("x" = 1:10, "y" = 1:10)
ggplot(df, aes(x, y)) +
  geom_point() +
  theme_minimal()
```



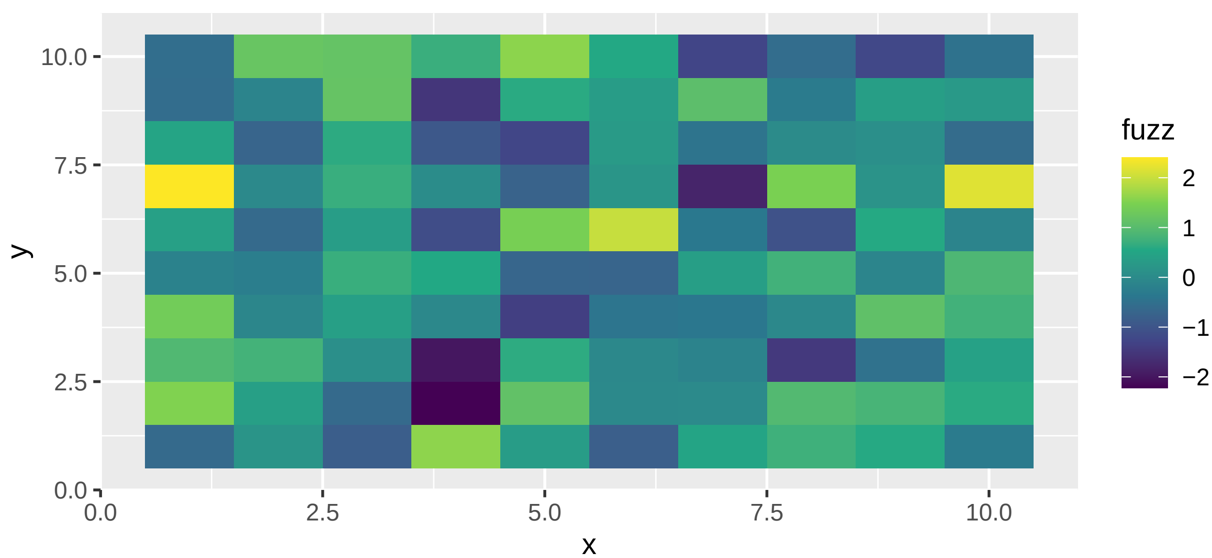
1b.

```
ggplot(df, aes(x, y)) +
  geom_point() +
  theme_minimal() +
  coord_equal()
```



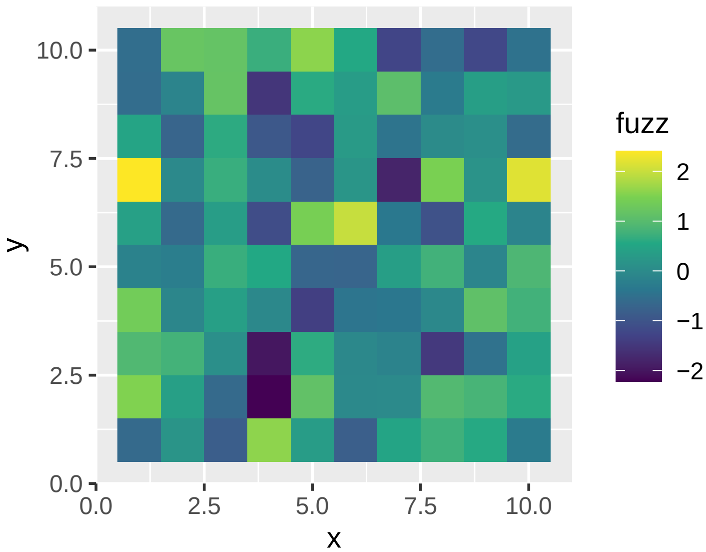
1c.

```r
set.seed(1)
fuzz <- rnorm(nrow(df))
ggplot(df,aes(x,y)) + geom_raster(aes(fill=fuzz))
```



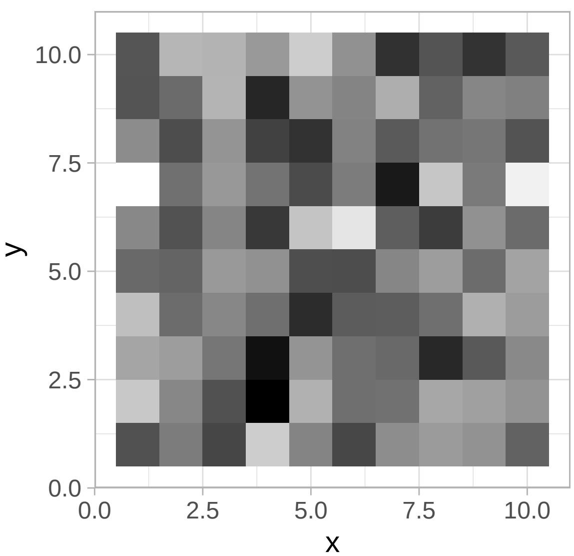
1d.

```r
ggplot(df,aes(x,y)) + geom_raster(aes(fill=fuzz)) + coord_equal()
```



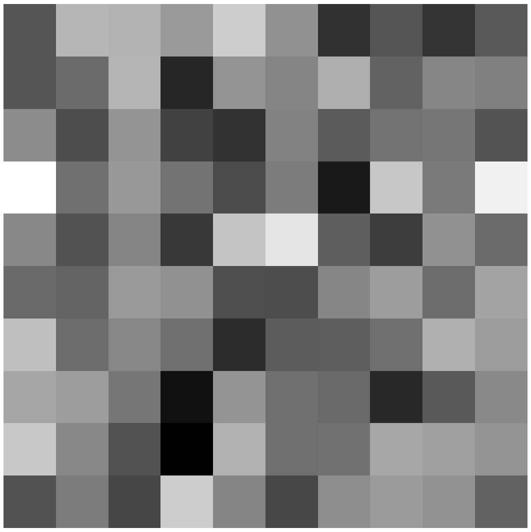
1e

```
#ggplot(df,aes(x,y)) + geom_raster(aes(fill=fuzz)) + scale_fill_gradient2(low="white", high="black") co
ggplot(df, aes(x = x, y = y)) +
  geom_raster(aes(fill = fuzz)) +
  scale_fill_gradient(low = "black", high = "white") +
  coord_equal() +
  theme_light() +
  theme(legend.position = "none")
```



1f.

```
ggplot(df, aes(x = x, y = y)) +
  geom_tile(aes(fill = fuzz)) +
  scale_fill_gradient(low = "black", high = "white") +
  coord_equal() +
  theme_void() +
  theme(legend.position = "none")
```



2a.

```
A <- matrix(c(-1,3,1,
              -7,9,1,
              -2,2,4), nrow = 3, byrow = TRUE)
ev <- eigen(A)
V <- ev$vectors
L <- ev$values
Lamda <- diag(L)
V_1 <- solve(V)
A
```

```
#      [,1] [,2] [,3]
# [1,]   -1    3    1
# [2,]   -7    9    1
# [3,]   -2    2    4
```

```
V %*% Lamda %*% V_1
```

```
#      [,1] [,2] [,3]
# [1,]   -1    3    1
# [2,]   -7    9    1
# [3,]   -2    2    4
```

2b.

```
A <- matrix(c(10,2,-6,2,7,0,-6,0,2), nrow = 3, byrow = TRUE)
ev <- eigen(A)
V <- ev$vectors
L <- ev$values
Lamda <- diag(L)
VT <-t(V)
A
```

```
#      [,1] [,2] [,3]
# [1,]   10    2   -6
# [2,]    2    7    0
# [3,]   -6    0    2
```

```r
zapsmall(V %*% Lamda %*% VT)
```

```
#      [,1] [,2] [,3]
# [1,]   10    2   -6
# [2,]    2    7    0
# [3,]   -6    0    2
```

```r
zapsmall(crossprod(V))
```

```
#      [,1] [,2] [,3]
# [1,]    1    0    0
# [2,]    0    1    0
# [3,]    0    0    1
```

2c.

```r
A <- matrix(c(1,2,3,4,5,6,7,8,6,8,10,12), nrow = 4, byrow = FALSE)
s_v_d <- svd(A)
U <- s_v_d$u
D <- s_v_d$d
E <- diag(D)
V <- s_v_d$v
A
```

```
#      [,1] [,2] [,3]
# [1,]    1    5    6
# [2,]    2    6    8
# [3,]    3    7   10
# [4,]    4    8   12
```

```r
zapsmall(U %*% E %*% t(V))
```

```
#      [,1] [,2] [,3]
# [1,]    1    5    6
# [2,]    2    6    8
# [3,]    3    7   10
# [4,]    4    8   12
```

2d/e

```r
A <- matrix(c(4,2,1,
              2,4,2,
              1,2,4), nrow = 3, byrow = TRUE)
ev <- eigen(A)
ev
```

```
# eigen() decomposition
# $values
# [1] 7.372281 3.000000 1.627719
#
# $vectors
#            [,1]          [,2]       [,3]
# [1,] -0.5417743 -7.071068e-01  0.4544013
# [2,] -0.6426206 -1.110223e-16 -0.7661846
# [3,] -0.5417743  7.071068e-01  0.4544013
```

```r
#you can do a Cholesky decomposition of A becasue A is positive definite since
#all of its eigenvalues are positive and it's symmetric.
```

```r
C <- chol(A)
C
```

```
#      [,1]     [,2]      [,3]
# [1,]    2 1.000000 0.5000000
# [2,]    0 1.732051 0.8660254
# [3,]    0 0.000000 1.7320508
```

```r
A
```

```
#      [,1] [,2] [,3]
# [1,]    4    2    1
# [2,]    2    4    2
# [3,]    1    2    4
```

```r
t(C) %*% C
```

```
#      [,1] [,2] [,3]
# [1,]    4    2    1
# [2,]    2    4    2
# [3,]    1    2    4
```

2f.

```r
A <- matrix(c(1,3,2,
              3,0,0,
              0,1,3,
              0,1,0), nrow = 4, byrow = TRUE)
QR <- qr(A)
R <- qr.R(QR)
Q <- qr.Q(QR)
X <- qr.X(QR)
X
```

```
#      [,1] [,2] [,3]
# [1,]    1    3    2
# [2,]    3    0    0
# [3,]    0    1    3
# [4,]    0    1    0
```

```r
zapsmall(Q %*% R)
```

```
#      [,1] [,2] [,3]
# [1,]    1    3    2
# [2,]    3    0    0
# [3,]    0    1    3
# [4,]    0    1    0
```