Assignment 3

```
personal_number <- 180
```

1.

(a)

Since U is a uniform on $[0,1]$, then $F_U(x) = x$

$$
\begin{aligned}
&F_{U-1}(x) = P[1-U \leq x] = 1 - P[1-U \geq x] = 1 - P[U \leq 1-x] = 1-(1-x) \\
&F_{U-1}(x) = x = F_U(x)
\end{aligned}
$$

And so, they have the same distribution

(b)

$$
\begin{aligned}
Var[e^U] &= \mathbb{E}[(e^U)^2] - \mathbb{E}[e^U]^2 \\
&= \int_0^1 e^{2U} dU - \left(\int_0^1 e^U dU\right)^2 \\
&= \frac{1}{2}(e^2-1) - (e-1)^2
\end{aligned}
$$

Since they are the same distribution, both U and $1-U$ have the same mean and variance.

(c)

$$
\begin{aligned}
Cov[e^U, e^{1-U}] &= \mathbb{E}[e^U e^{1-U}] - \mathbb{E}[e^U]\mathbb{E}[e^{1-U}] \\
&= \mathbb{E}[e^{U+1-U}] - \left(\int_0^1 e^U dU\right)\left(\int_0^1 e^{1-U} dU\right) \\
&= e - (e-1)(e^2)(-e^{-1}+1)
\end{aligned}
$$

(d)

$$
\begin{aligned}
Var[e^U + e^{U-1}] &= Var[e^U] + Var[e^{1-U}] + 2Cov(e^U, e^{1-U}) \\
&= 2(\frac{1}{2}(e^2-1) - (e-1)^2 + e - (e-1)(e^2)(-e^{-1}+1))
\end{aligned}
$$

(e)

$$
\begin{aligned}
\frac{Var[e^U + e^{1-U}]}{Var[e^U] + Var[e^{1-U}]} &= \frac{2(\frac{1}{2}(e^2-1) - (e-1)^2 + e - (e-1)(e^2)(-e^{-1}+1))}{2(\frac{1}{2}(e^2-1) - (e-1)^2)} \\
&\approx -20.9\overline{28}
\end{aligned}
$$

2.

(a)

$$
\begin{aligned}
Cor(aX+b, cY+d) &= \frac{Cov(aX+b, cY+d)}{\sqrt{Var(aX+b)Var(cY+d)}} \\
&= \frac{Cov(aX, cY)}{\sqrt{Var(aX)Var(cY)}} \\
&= \frac{acCov(X,Y)}{\sqrt{a^2c^2Var(X)Var(Y)}} = \frac{acCov(X,Y)}{\sqrt{(ac)^2\sqrt{Var(X)Var(Y)}}} \\
&= \frac{acCov(X,Y)}{ac\sqrt{Var(X)Var(Y)}} = \frac{Cov(X,Y)}{\sqrt{Var(X)Var(Y)}} \\
&= Cor(X,Y)
\end{aligned}
$$

(b)

Using seed 007:

```
g2 = function (x) exp(-x)/(1+x^2)
h2 = function (x) 1 - x
set.seed(007)
u = runif(10000)

cor(g2(u), h2(u))
```

```
## [1] 0.9905712
```

```
A = -cov(g2(u),h2(u))/var(h2(u)) ;A
```

```
## [1] -0.8407234
```

```
u = runif(100000)

mean(g2(u))
```

```
## [1] 0.5253908
```

```
mean(g2(u) + A*(h2(u) - 0.5))
```

```
## [1] 0.5249112
```

```
v1=var(g2(u)) ; v1
```

```
## [1] 0.06027686
```

```r
v2 = var(g2(u) + A*(h2(u) - 0.5)); v2
```

```
## [1] 0.001115375
```

```r
(v1-v2)/v1
```

```
## [1] 0.9814958
```

Using my personal seed 180:

```r
g2 = function (x) exp(-x)/(1+x^2)
h2 = function (x) 1 - x
set.seed(personal_number)
u = runif(10000)

cor(g2(u), h2(u))
```

```
## [1] 0.9908035
```

```r
A = -cov(g2(u),h2(u))/var(h2(u)) ;A
```

```
## [1] -0.8407016
```

```r
u = runif(100000)

mean(g2(u))
```

```
## [1] 0.5233347
```

```r
mean(g2(u) + A*(h2(u) - 0.5))
```

```
## [1] 0.5248801
```

```r
v1=var(g2(u)) ; v1
```

```
## [1] 0.0600959
```

```r
v2 = var(g2(u) + A*(h2(u) - 0.5)); v2
```

```
## [1] 0.001115457
```

```r
(v1-v2)/v1
```

```
## [1] 0.9814387
```

```r
# Now for f3

h3 = function (x) 1 - (0.9*x)
set.seed(180)
u = runif(10000)

cor(g2(u), h3(u))
```

```
## [1] 0.9908035
```

```r
A = -cov(g2(u),h3(u))/var(h3(u)) ;A
```

```
## [1] -0.9341129
```

```r
u = runif(100000)

mean(g2(u))
```

```
## [1] 0.5233347
```

```r
mean(g2(u) + A*(h3(u) - 0.5))
```

```
## [1] 0.4781745
```

```r
v1=var(g2(u)) ; v1
```

```
## [1] 0.0600959
```

```r
v2 = var(g2(u) + A*(h3(u) - 0.5)); v2
```

```
## [1] 0.001115457
```

```r
(v1-v2)/v1
```

```
## [1] 0.9814387
```

From here, we can see that f_3 is performing about the same as our f_2.

(c)

This might be because our f_2 and f_3 are extremely similar, and so scaling our x isn't enough to cause a difference in our results.

3.

```r
set.seed(personal_number)
n <- 10000
sum(rnorm(n) > 4.5)
```

```
## [1] 0
```

```r
sum(rnorm(n) > 4.5)
```

```
## [1] 0
```

```r
sum(rnorm(n) > 4.5)
```

```
## [1] 0
```

```r
sum(rnorm(n) > 4.5)
```

```
## [1] 0
```

```r
n <- 100000
sum(rnorm(n) > 4.5)
```

```
## [1] 0
```

```r
n <- 1000000
sum(rnorm(n) > 4.5)
```

```
## [1] 4
```

Based on the result above, we can see that getting a value over 4.5 is incredibly low. When running the program multiple times and collecting multiple samples, we are not getting many cases. We only seem to get some cases of values over 4.5 when we set N=1 000 000.

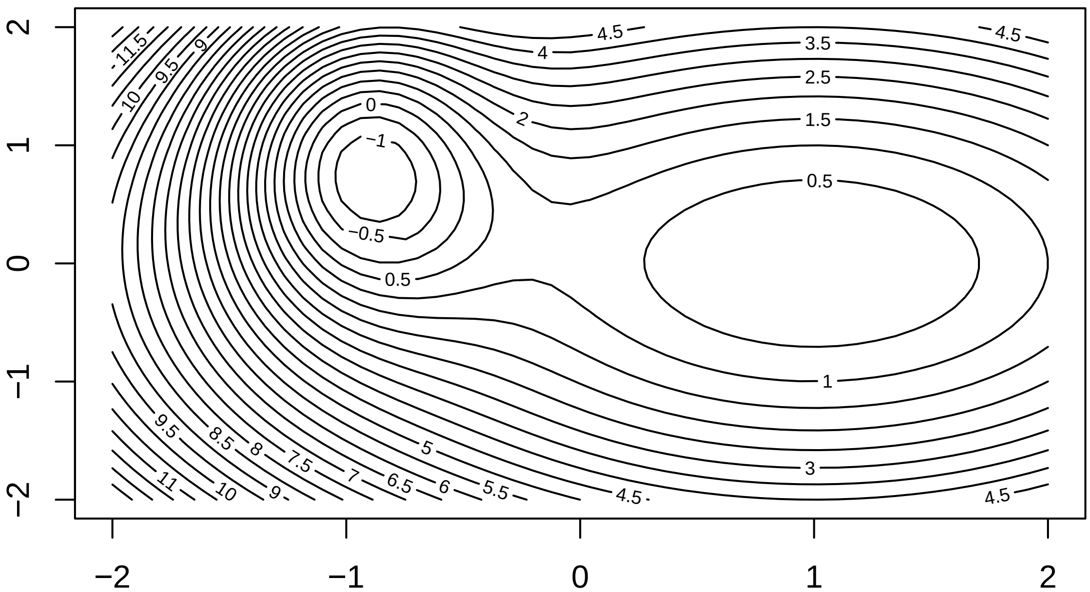
4.

(a)

```r
# Define the grid
x = seq(-2, 2, len = 50)
y = x
xy = expand.grid(x, y)

e = -6*exp(-(3*xy[,1]^2 + 0.5*xy[,2]^2 + 6*xy[,1] - xy[,2] + 3.5))
p = xy[,1]^2 + xy[,2]^2 -2*xy[,1] + 1
z = matrix(e + p, nrow=length(x), ncol=length(y))

contour(x, y, z, nlevels = 50)
```



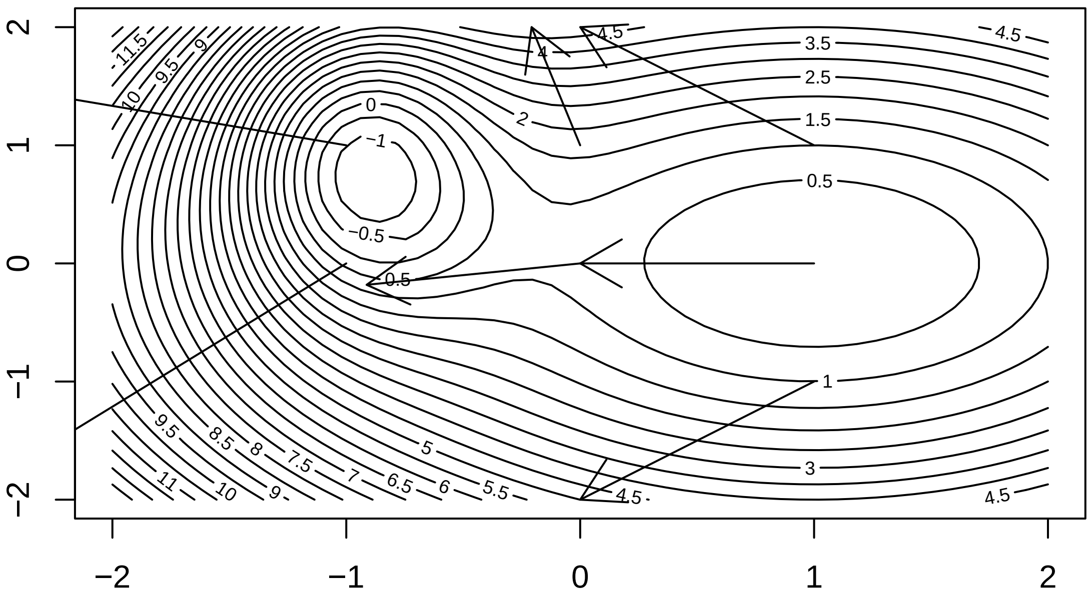
(b)

$$D_x f(x,y) = \frac{\partial}{\partial x} - 6e^{-(3x^2+0.5y^2+6x-y+3.5)} + x^2 + y^2 - 2x + 1$$
$$= (6e^{-(3x^2+0.5y^2+6x-y+3.5)})(6x+6) + 2x - 2$$
$$D_y f(x,y) = \frac{\partial}{\partial y} - 6e^{-(3x^2+0.5y^2+6x-y+3.5)} + x^2 + y^2 - 2x + 1$$
$$= (6e^{-(3x^2+0.5y^2+6x-y+3.5)})(y-1) + 2y$$
$$\therefore \nabla f(x,y) = \begin{pmatrix} (6e^{-(3x^2+0.5y^2+6x-y+3.5)})(6x+6) + 2x - 2 \\ (6e^{-(3x^2+0.5y^2+6x-y+3.5)})(y-1) + 2y \end{pmatrix}$$

```
d_x = function(x, y) {
  exp_term <- exp(-(3*x^2 + 0.5*y^2 + 6*x - y + 3.5))
  return((6 * exp_term) * (6*x + 6) + 2*x - 2)
}

d_y = function(x, y) {
  exp_term <- exp(-(3*x^2 + 0.5*y^2 + 6*x - y + 3.5))
  return((6 * exp_term) * (y - 1) + 2*y)
}

contour(x, y, z, nlevels = 50)
arrows(0,0,d_x(0,0),d_y(0,0))
arrows(1,1,d_x(1,1),d_y(1,1))
arrows(-1,0,d_x(-1,0),d_y(-1,0))
arrows(1,0,d_x(1,0),d_y(1,0))
arrows(0,1,d_x(0,1),d_y(0,1))
arrows(1,-1,d_x(1,-1),d_y(1,-1))
arrows(-1,1,d_x(-1,1),d_y(-1,1))
```



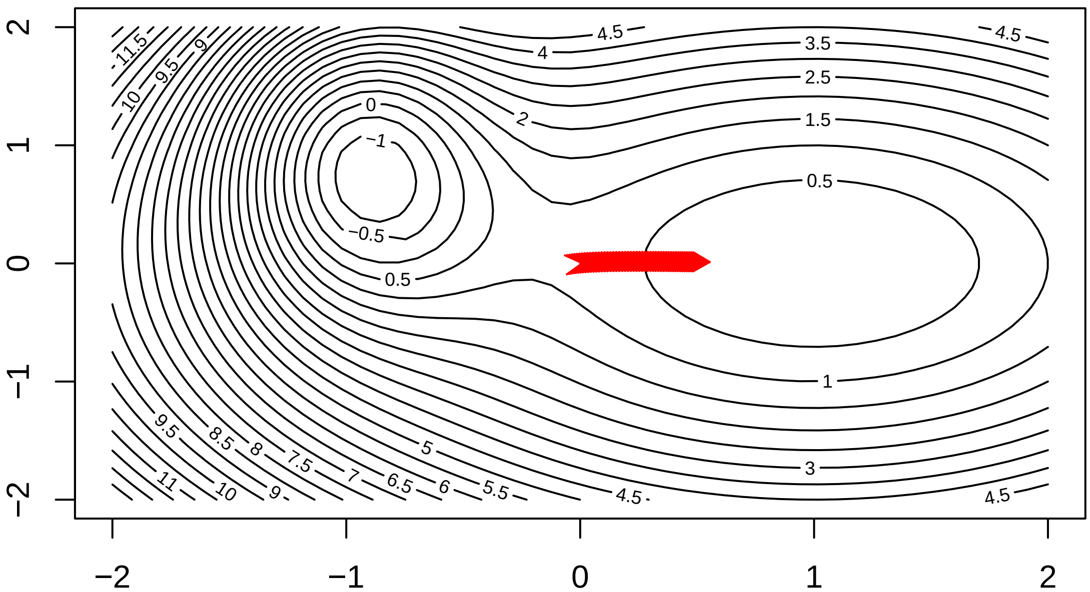
(c)

```r
f = function(x,y) {
  exp_t = 6*exp(-(3*x^2 + 0.5*y^2 + 6*x -y +3.5))
  return(-exp_t + x^2 + y^2 -2*x + 1)
}
gradient = function(x,y) {
  return(c(d_x(x, y), d_y(x, y)))
}

steps <- list(c(0,0))

for (i in 1:50) {
  steps[[i + 1]] = steps[[i]] - 0.01 * gradient(steps[[i]][1], steps[[i]][2])
}

contour(x, y, z, nlevels = 50)
for (i in 1:(length(steps) - 1)) {
  arrows(steps[[i]][1], steps[[i]][2], steps[[i + 1]][1], steps[[i + 1]][2], col = "red", length = 0.1)
}
```



(d)

So it seems as though that the gradient descent algorithm goes towards the peak of our contour lines, however it only goes towards any stationary point of a function. In our case, we see that the gradient descent algorithm actually is converging to a local minimum, but not the global minimum (which we can observed based on contour values).