Econ 403A: Homework 2

Minxuan Wang 705060599

October 24, 2017

1 Generate Plots

a.I choose the 1980-2017 GDP, Unemployment Rate and Inflation Rate(by CPI) annual data of the US, China and Japan, put them in a single csv document.

```
setwd("D:/Econ 403A/Homework 2")
data1<-read.csv("countrydata.csv")
data1
         Date       GDP   LUR     CPI Country
1   2016/12/31 18569.100 4.850 242.821      US
2   2015/12/31 18036.650 5.258 237.846      US
3   2014/12/31 17393.100 6.167 236.290      US
4   2013/12/31 16691.500 7.367 234.723      US
5   2012/12/31 16155.250 8.075 231.221      US
6   2011/12/31 15517.925 8.933 227.223      US
7   2010/12/31 14964.400 9.608 220.472      US
8   2009/12/31 14418.725 9.283 217.347      US
9   2008/12/31 14718.575 5.800 211.398      US
10  2007/12/31 14477.625 4.617 211.445      US
11  2006/12/31 13855.900 4.608 203.100      US
12  2005/12/31 13093.700 5.083 198.100      US
13  2004/12/31 12274.925 5.542 191.700      US
14  2003/12/31 11510.675 5.992 185.500      US
15  2002/12/31 10977.525 5.783 181.800      US
16  2001/12/31 10621.825 4.742 177.400      US
17  2000/12/31 10284.750 3.967 174.600      US
18  1999/12/31  9660.625 4.217 168.800      US
19  1998/12/31  9089.150 4.500 164.400      US
20  1997/12/31  8608.525 4.942 161.800      US
21  1996/12/31  8100.175 5.408 159.100      US
22  1995/12/31  7664.050 5.592 153.900      US
23  1994/12/31  7308.775 6.100 150.100      US
24  1993/12/31  6878.700 6.908 146.300      US
25  1992/12/31  6539.300 7.492 142.300      US
26  1991/12/31  6174.050 6.850 138.200      US
```

27	1990/12/31	5979.575	5.617	134.200	US
28	1989/12/31	5657.700	5.258	126.300	US
29	1988/12/31	5252.625	5.492	120.700	US
30	1987/12/31	4870.225	6.175	115.600	US
31	1986/12/31	4590.125	7.000	110.800	US
32	1985/12/31	4346.750	7.192	109.500	US
33	1984/12/31	4040.700	7.508	105.500	US
34	1983/12/31	3638.125	9.600	101.400	US
35	1982/12/31	3345.000	9.708	97.700	US
36	1981/12/31	3210.950	7.617	94.100	US
37	1980/12/31	2862.475	7.175	86.400	US
38	2016/12/31	11218.281	4.020	117.168	CHN
39	2015/12/31	11226.186	4.050	114.867	CHN
40	2014/12/31	10534.526	4.090	113.236	CHN
41	2013/12/31	9635.025	4.050	111.028	CHN
42	2012/12/31	8570.348	4.090	108.189	CHN
43	2011/12/31	7522.103	4.090	105.400	CHN
44	2010/12/31	6066.351	4.140	100.000	CHN
45	2009/12/31	5121.681	4.300	96.805	CHN
46	2008/12/31	4604.285	4.200	97.488	CHN
47	2007/12/31	3571.451	4.000	92.057	CHN
48	2006/12/31	2774.308	4.100	87.840	CHN
49	2005/12/31	2308.786	4.200	86.542	CHN
50	2004/12/31	1966.223	4.200	85.012	CHN
51	2003/12/31	1671.072	4.300	81.821	CHN
52	2002/12/31	1477.483	4.000	80.851	CHN
53	2001/12/31	1344.097	3.600	81.503	CHN
54	2000/12/31	1214.912	3.100	80.936	CHN
55	1999/12/31	1097.133	3.100	80.614	CHN
56	1998/12/31	1032.576	3.100	81.758	CHN
57	1997/12/31	965.320	3.100	82.418	CHN
58	1996/12/31	867.224	3.000	80.173	CHN
59	1995/12/31	736.870	2.900	74.028	CHN
60	1994/12/31	566.471	2.800	63.218	CHN
61	1993/12/31	623.054	2.600	50.941	CHN
62	1992/12/31	495.671	2.300	44.413	CHN
63	1991/12/31	415.604	2.300	41.741	CHN
64	1990/12/31	398.623	2.500	40.369	CHN
65	1989/12/31	461.066	2.600	39.155	CHN
66	1988/12/31	411.923	2.000	33.182	CHN
67	1987/12/31	330.303	2.000	27.931	CHN
68	1986/12/31	303.340	2.000	26.031	CHN
69	1985/12/31	312.616	1.800	24.442	CHN
70	1984/12/31	316.666	1.900	22.362	CHN
71	1983/12/31	307.683	2.300	21.774	CHN
72	1982/12/31	286.729	3.200	21.347	CHN

```
73  1981/12/31   290.724 3.800  20.929     CHN
74  1980/12/31   305.350 4.900  20.418     CHN
75  2016/12/31  4938.644 3.108 100.100     JPN
76  2015/12/31  4382.420 3.375  99.800     JPN
77  2014/12/31  4848.733 3.583  99.700     JPN
78  2013/12/31  5155.716 4.008  97.400     JPN
79  2012/12/31  6203.213 4.325  95.800     JPN
80  2011/12/31  6157.460 4.583  96.000     JPN
81  2010/12/31  5700.099 5.058  96.200     JPN
82  2009/12/31  5231.384 5.075  96.500     JPN
83  2008/12/31  5037.910 3.983  98.200     JPN
84  2007/12/31  4515.264 3.833  97.800     JPN
85  2006/12/31  4530.475 4.117  97.100     JPN
86  2005/12/31  4755.980 4.425  96.800     JPN
87  2004/12/31  4815.772 4.733  97.200     JPN
88  2003/12/31  4447.378 5.242  97.000     JPN
89  2002/12/31  4115.197 5.358  97.400     JPN
90  2001/12/31  4304.758 5.042  97.700     JPN
91  2000/12/31  4887.301 4.733  98.900     JPN
92  1999/12/31  4546.050 4.667  99.300     JPN
93  1998/12/31  4034.448 4.100 100.400     JPN
94  1997/12/31  4415.715 3.400  99.800     JPN
95  1996/12/31  4834.019 3.367  98.000     JPN
96  1995/12/31  5450.805 3.150  97.400     JPN
97  1994/12/31  4907.582 2.892  97.800     JPN
98  1993/12/31  4467.123 2.500  97.200     JPN
99  1992/12/31  3898.138 2.150  96.200     JPN
100 1991/12/31  3582.802 2.092  95.100     JPN
101 1990/12/31  3140.669 2.100  92.600     JPN
102 1989/12/31  3052.896 2.250  89.300     JPN
103 1988/12/31  3051.174 2.517  87.000     JPN
104 1987/12/31  2514.969 2.850  89.200     JPN
105 1986/12/31  2075.616 2.767  88.500     JPN
106 1985/12/31  1401.006 2.625  88.800     JPN
107 1984/12/31  1309.788 2.708  87.500     JPN
108 1983/12/31  1232.408 2.658  85.200     JPN
109 1982/12/31  1130.404 2.350  83.800     JPN
110 1981/12/31  1215.778 2.208  82.100     JPN
111 1980/12/31  1099.695 2.017  78.800     JPN
```

First of all, we should define the function of ”marginal_ plot” (https://github.com/ChrKoenig/R_ marginal_ plot).

```
marginal_plot = function(x, y, group = NULL, data = NULL,
lm_show = FALSE, lm_formula = y ~ x, bw = "nrd0", adjust = 1,
alpha = 1, plot_legend = T, ...){
require(scales)
```

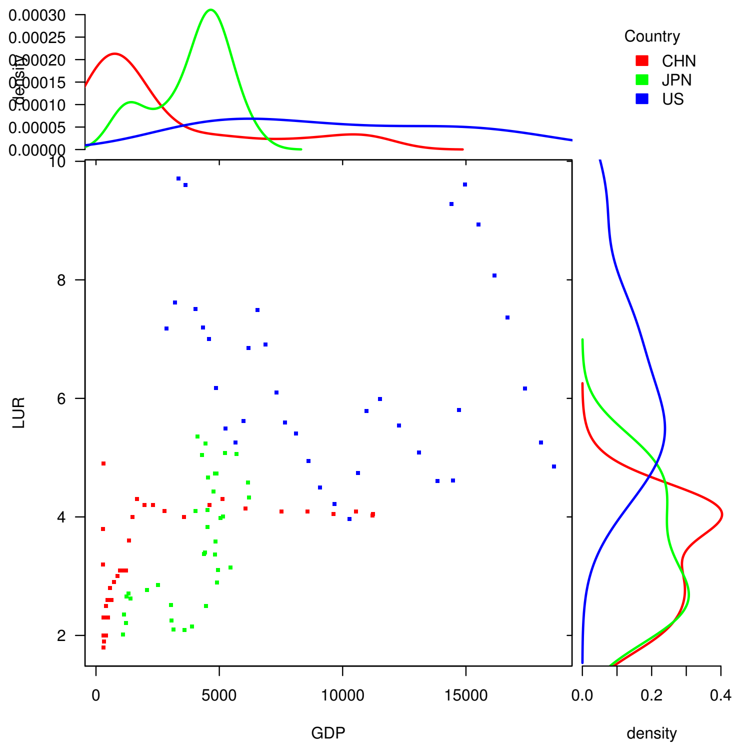
```
moreargs = eval(substitute(list(...)))

# prepare consistent df
if(missing(group)){
if(missing(data)){
if(length(x) != length(y)){stop("Length of arguments not equal")}
data = data.frame(x = as.numeric(x), y = as.numeric(y))
} else {
data = data.frame(x = as.numeric(data[,deparse(substitute(x))]),
                  y = as.numeric(data[,deparse(substitute(y))]))
}
if(sum(!complete.cases(data)) > 0){
warning(sprintf("Removed %i rows with missing data",
sum(!complete.cases(data))))
data = data[complete.cases(data),]
}
group_colors = "black"
 } else {
if(missing(data)){
if(length(x) != length(y) | length(x) != length(group))
{stop("Length of arguments not equal")}
data = data.frame(x = as.numeric(x), y = as.numeric(y),
group = as.factor(group))
} else {
data = data.frame(x = as.numeric(data[,deparse(substitute(x))]),
                  y = as.numeric(data[,deparse(substitute(y))]),
          group = as.factor(data[,deparse(substitute(group))]))
}
if(sum(!complete.cases(data)) > 0){
warning(sprintf("Removed %i rows with missing data",
sum(!complete.cases(data))))
data = data[complete.cases(data),]
}
data = subset(data, group %in% names(which(table(data$group)>5)))
data$group = droplevels(data$group)
group_colors = rainbow(length(unique(data$group)))
}

# log-transform data (this is need for correct plotting of
  density functions)
if(!is.null(moreargs$log)){
if(!moreargs$log %in% c("y", "x", "yx", "xy")){
warning("Ignoring invalid 'log' argument. Use'y','x','yx'or'xy.")
} else {
data = data[apply(data[unlist(strsplit(moreargs$log, ""))], 1,
function(x) !any(x <= 0)), ]
```

```
data[,unlist(strsplit(moreargs$log, ""))] = log10(data
[,unlist(strsplit(moreargs$log, ""))])
}
moreargs$log = NULL
# remove to prevent double logarithm when plotting
}

# Catch unwanted user inputs
if(!is.null(moreargs$col)){moreargs$col = NULL}
if(!is.null(moreargs$type)){moreargs$type = "p"}

# get some default plotting arguments
if(is.null(moreargs$xlim)){moreargs$xlim = range(data$x)}
if(is.null(moreargs$ylim)){moreargs$ylim = range(data$y)}
if(is.null(moreargs$xlab)){moreargs$xlab = deparse(substitute(x))}
if(is.null(moreargs$ylab)){moreargs$ylab = deparse(substitute(y))}
if(is.null(moreargs$las)){moreargs$las = 1}

# plotting
tryCatch(expr = {
ifelse(!is.null(data$group),
data_split <- split(data, data$group), data_split <- list(data))
orig_par = par(no.readonly = T)
par(mar = c(0.25,5,1,0))
layout(matrix(1:4, nrow = 2, byrow = T),
widths = c(10,3), heights = c(3,10))

# upper density plot
plot(NULL, type = "n", xlim = moreargs$xlim, ylab = "density",
ylim = c(0, max(sapply(data_split, function(group_set)
max(density(group_set$x, bw = bw)$y)))), main = NA, axes = F)
axis(2, las = 1)
mapply(function(group_set, group_color)
{lines(density(group_set$x, bw = bw, adjust = adjust),
col = group_color, lwd = 2)}, data_split, group_colors)

# legend
par(mar = c(0.25,0.25,0,0))
plot.new()
if(!missing(group) & plot_legend){
legend("center", levels(data$group), fill=group_colors,
border = group_colors, bty = "n",title = deparse(substitute(group)),
title.adj=0.1)
}

# main plot
```

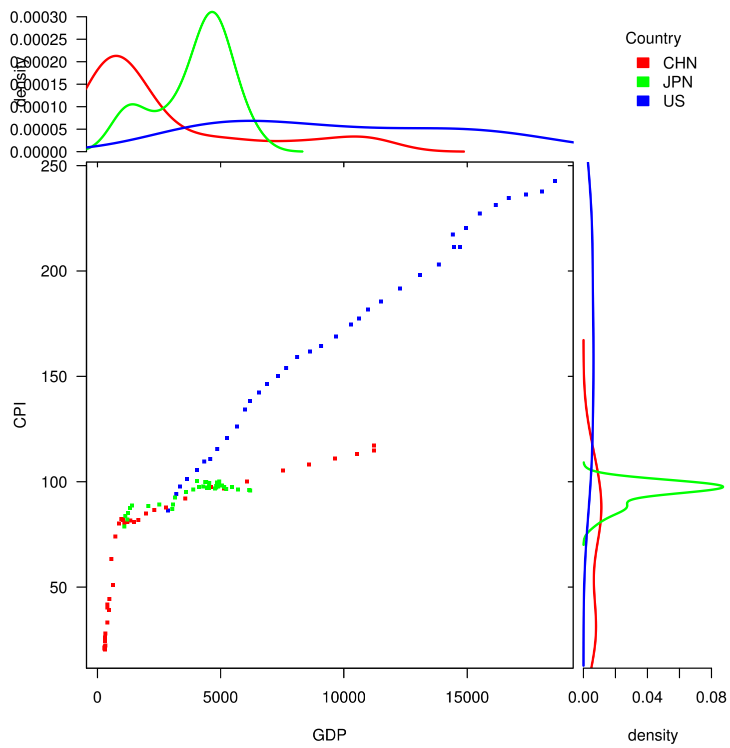
```
par(mar = c(4,5,0,0))
if(missing(group)){
do.call(plot, c(list(x = quote(data$x), y = quote(data$y), col =
quote(scales::alpha("black", alpha))), moreargs))
} else {
do.call(plot, c(list(x = quote(data$x), y = quote(data$y), col =
quote(scales::alpha(group_colors[data$group], alpha))), moreargs))
}
axis(3, labels = F, tck = 0.01)
axis(4, labels = F, tck = 0.01)
box()

if(lm_show == TRUE & !is.null(lm_formula)){
mapply(function(group_set, group_color){
lm_tmp = lm(lm_formula, data = group_set)
x_coords = seq(min(group_set$x), max(group_set$x), length.out=100)
y_coords = predict(lm_tmp, newdata = data.frame(x = x_coords))
lines(x = x_coords, y = y_coords, col = group_color, lwd = 2.5)
}, data_split,
rgb(t(ceiling(col2rgb(group_colors)*0.8)), maxColorValue = 255))
}
# right density plot
par(mar = c(4,0.25,0,1))
plot(NULL, type = "n", ylim = moreargs$ylim, xlim = c(0,
max(sapply(data_split, function(group_set)max(density
(group_set$y,bw = bw)$y)))), main = NA, axes = F, xlab = "density")
mapply(function(group_set, group_color)
{lines(x = density(group_set$y, bw = bw, adjust = adjust)$y, y =
density(group_set$y, bw = bw)$x,
col = group_color, lwd = 2)}, data_split, group_colors)
axis(1)
}, finally = {
par(orig_par)
})
}
```

Generate the scatter plot with marginal densities.

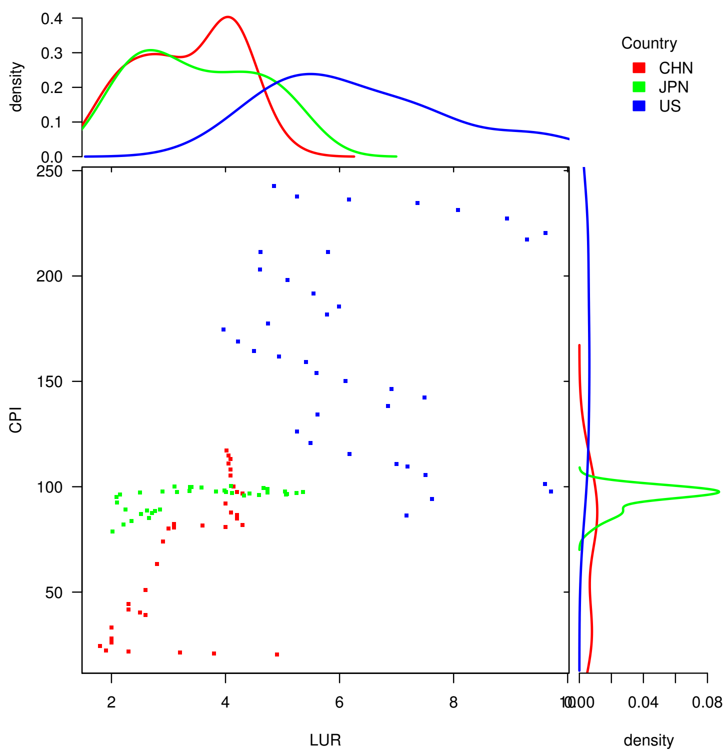
```
# GDP with Unemployment rate
marginal_plot(x=GDP,y=LUR,group=Country,data=data1,bw="nrd",
xlab="GDP",ylab="LUR",pch=15,cex=0.5)
```




```
# GDP with Inflation
marginal_plot(x=GDP,y=CPI,group=Country,data=data1,bw="nrd",
xlab="GDP",ylab="CPI",pch=15,cex=0.5)
```

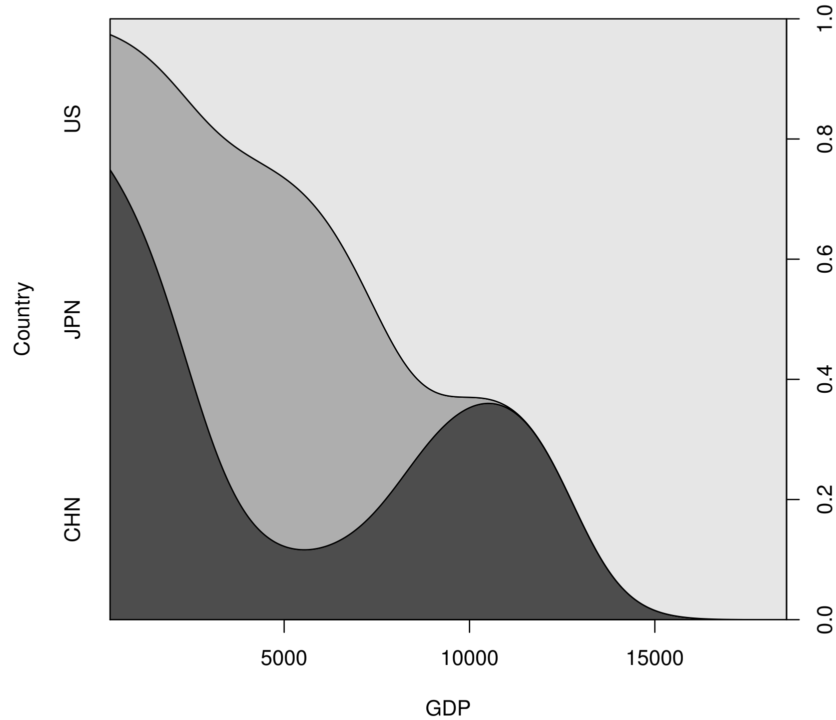



```
# Unemployment rate with Inflation
marginal_plot(x=LUR,y=CPI,group=Country,data=data1,bw="nrd",
xlab="LUR",ylab="CPI",pch=15,cex=0.5)
```

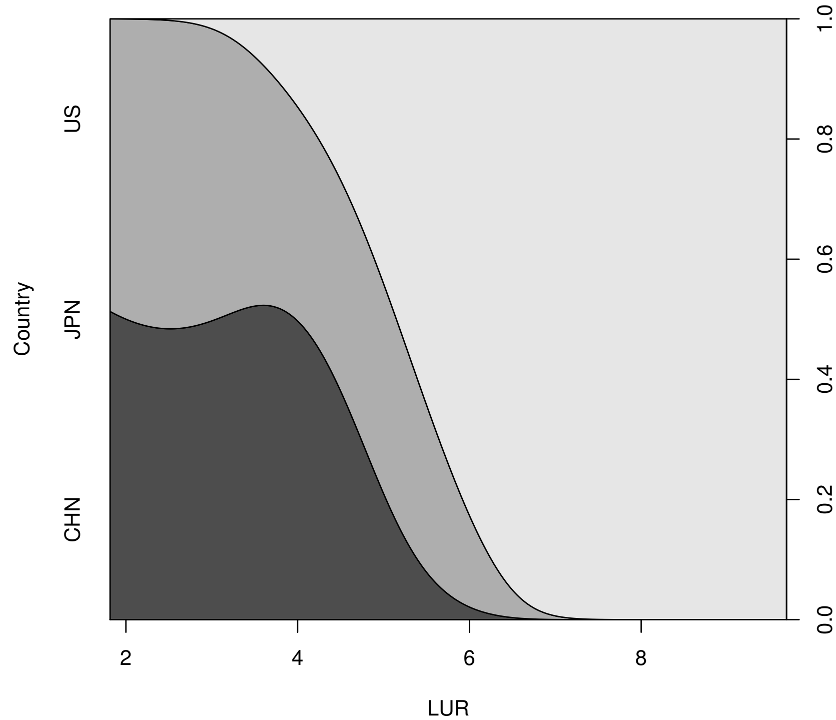


b.I will choose the GDP, LUR and CPI of 2016 as an example:

```
# Conditional Distribution of GDP and Country
data1[c(1,38,75),]
         Date       GDP   LUR     CPI Country
1  2016/12/31 18569.100 4.850 242.821      US
38 2016/12/31 11218.281 4.020 117.168     CHN
75 2016/12/31  4938.644 3.108 100.100     JPN

cdplot(Country~GDP,data=data1)
```




```
# Conditional Distribution of LUR and Country
data1[c(1,38,75),]
         Date       GDP   LUR     CPI Country
1  2016/12/31 18569.100 4.850 242.821      US
38 2016/12/31 11218.281 4.020 117.168     CHN
75 2016/12/31  4938.644 3.108 100.100     JPN

cdplot(Country~LUR,data=data1)
```



2 X and Y have Joint Density

$f(xy) = cx(y-x)e^{-y}, 0 \leq x \leq y \leq 1$

a.Find c

$\int_0^\infty \int_x^\infty cx(y-x)e^{-y}dxdy = 1$

$= \int_0^\infty cx \int_x^\infty (y-x)e^{-y}dydx$

$= \int_0^\infty cx \left[\int_x^\infty -(y-x)d(e^{-y})\right] dx$

$= \int_0^\infty cx \left[-\left[(y-x)e^{-y}|_x^\infty - \int_x^\infty e^{-y}dy\right]\right] dx$

$= \int_0^\infty cx \left[-e^{-y}|_x^\infty\right] dx$

$= \int_0^\infty cx \left[0 - (-e^{-x})\right] dx$

$= \int_0^\infty cxe^{-x}dx$

$= -c\int_0^\infty xd(e^{-x})$

$= -c\left[xe^{-x}|_0^\infty - \int_0^\infty e^{-x}dx\right]$

$= -c\left[0 - (-e^{-x})|_0^\infty\right]$

$= -c(0-1)$

$= c = 1$

So c=1

* When calculating $\lim\limits_{y\to\infty} \frac{y-x}{e^y}$, we should use L'Hopital's rule that it equals to $\lim\limits_{y\to\infty} \frac{1}{e^y} \to 0$. We will use this rule many times later.

b.Find the marginal density function

$f_X(x) = \int_x^\infty f(x,y)dy$

$= \int_x^\infty x(y-x)e^{-y}dy$

$= \int_x^\infty \left[-x(y-x)\right]d(e^{-y})$

$= \int_x^\infty (x-y)d(e^{-y})$

$= (x-y)xe^{-y}|_x^\infty - \int_x^\infty e^{-y}dx(x-y)$

$= \frac{x(x-y)}{e^y}|_x^\infty - \int_x^\infty (-xe^{-y})dy$

$= -\left(\frac{x}{e^y}|_x^\infty\right)$

$= xe^{-x}$

$f_Y(y) = \int_0^y f(x,y)dx$

$= \int_0^y x(y-x)e^{-y}dx$

$= \frac{1}{2}ye^{-y}x^2 - \frac{1}{3}e^{-y}x^3|_0^y$

$= \frac{1}{2}y^3e^{-y} - \frac{1}{3}e^{-y}y^3$

$= \frac{1}{6}y^3e^{-y}$

$$f_{X|Y}(x|y) = \frac{f(x,y)}{f_Y y} = \frac{x(y-x)e^{-y}}{\frac{1}{6}y^3 e^{-y}} = \frac{6x(y-x)}{y^3}$$
$$f_{Y|X}(y|x) = \frac{f(x,y)}{f_X x} = \frac{x(y-x)e^{-y}}{xe^{-x}} = \frac{e^x(y-x)}{e^y}$$

c.Find the conditional expectation function

$E(X|Y=y) = \int_0^y x f_{X|Y}(x|y)dx$

$= \int_0^y 6\frac{x^2(y-x)}{y^3}dx$

$= \int_0^y \left(\frac{6}{y^2}x^2 - \frac{6}{y^3}x^3\right)dx$

$= \left(\frac{2}{y^2}x^3 - \frac{3}{2y^3}x^4\right)|_0^y$

$= 2y - \frac{3y}{2} = \frac{1}{2}y$

$E(X|Y) = \frac{1}{2}Y$

$E(Y|X=x) = \int_x^\infty ye^x(y-x)e^{-y}dy$

$= e^x\int_x^\infty y(y-x)e^{-y}dy$

$= e^x\int_x^\infty \left(y^2e^{-y} - yxe^{-y}\right)dy$

$= e^x\left[-\int_x^\infty y^2 d\left(e^{-y}\right) - x\int_x^\infty ye^{-y}dy\right]$

$= e^x\left[-e^{-y}y^2|_x^\infty + \int_x^\infty 2ye^{-y}dy + x\int_x^\infty yd(e^{-y}\right]$

$= e^x\left[e^{-x}x^2 - 2\int_x^\infty yd(e^{-y}) + x\int_x^\infty yd(e^{-y})\right]$

$= e^x\left[e^{-x}x^2 + (x-2)\left(ye^{-y}|_x^\infty - \int_x^\infty e^{-y}dy\right)\right]$

$= e^x\left[x^2e^{-x} + (x-2)\left(-xe^{-x} - e^{-x}\right)\right]$

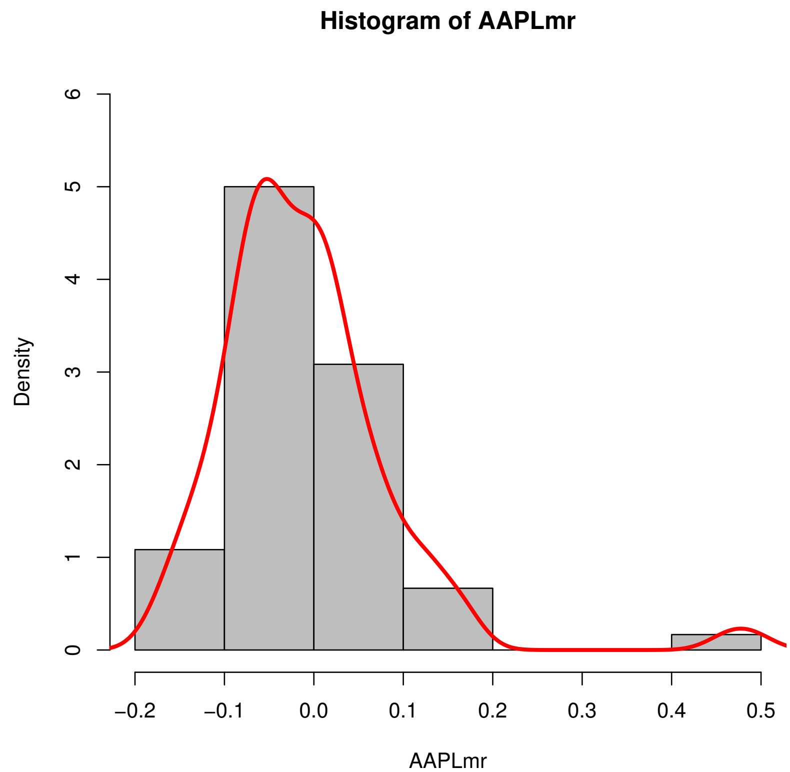
$= x^2 + (x-2)(-x-1) = x+2$

$E(Y|X) = X+2$

3 R Exercise

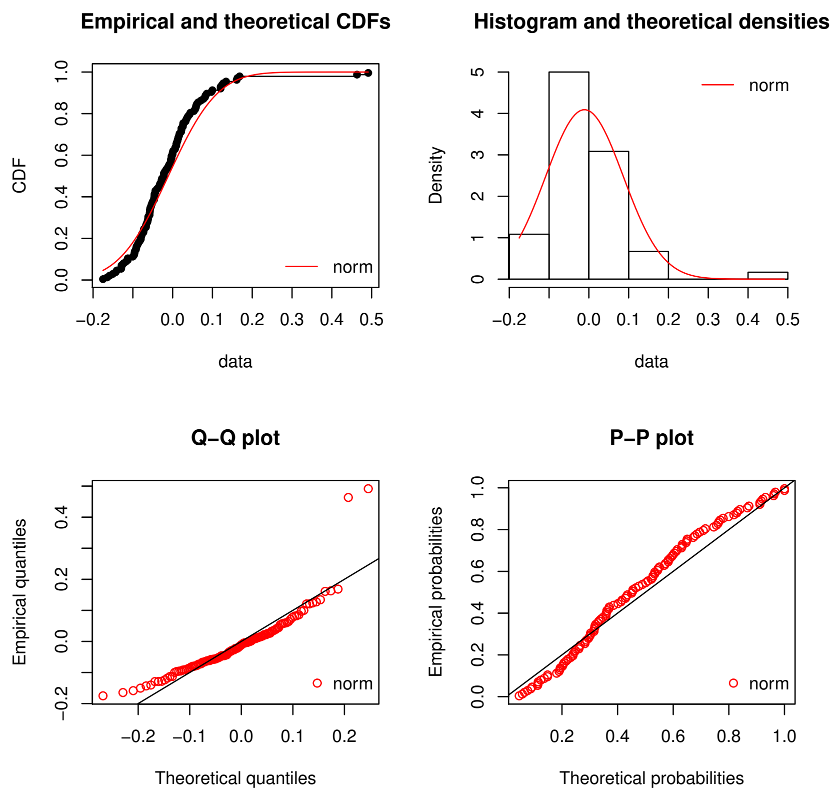
I choose the stock of Apple, since 2007-10-19 to 2017-10-19.

```
# Use the two following functions
library(Quandl)
library(quantmod)
# Collect data from Quandl
data<-Quandl("EOD/AAPL", api_key="QR_sKtW9xGJddDnfzCbT",
collapse="monthly", start_date="2007-10-19")
AAPL<-data$Adj_Close
# Calculate the daily return
AAPLmr<-Delt(AAPL)
# Remove the N/A value
AAPLmr<-AAPLmr[!is.na(AAPLmr)]
AAPLmr
  [1] -1.363200e-02  6.410589e-02 -9.677337e-02 -3.166812e-02  6.068602e-02
  [6] -6.346854e-02  6.961364e-05 -4.642907e-02 -1.179656e-01 -4.557066e-02
 [11] -4.576066e-02  2.202125e-02 -4.315660e-03 -6.147722e-02 -2.307312e-02
 [16] -8.262163e-02  4.456067e-02 -6.698953e-02  1.626840e-01 -1.128544e-01
 [21]  1.332321e-03  8.136429e-02  1.238837e-01  5.818318e-03 -7.991632e-02
 [26]  2.555707e-02  7.089922e-02  3.400660e-02  3.870839e-02 -4.334814e-02
 [31] -5.753096e-03  3.238769e-02 -9.152510e-02 -5.786958e-02  7.745968e-02
 [36] -9.581235e-02 -6.712963e-02  1.736973e-02 -7.193383e-02 -2.792887e-02
 [41] -2.691734e-02 -7.297532e-02 -9.040994e-02 -1.956254e-02 -5.435067e-02
 [46]  1.206952e-01 -8.823215e-03 -6.547848e-02 -8.791242e-02  2.195281e-02
 [51] -7.729676e-02 -1.237487e-01  1.341765e-01 -2.199512e-02 -2.710150e-04
 [56] -2.846428e-03  2.611363e-02  1.683525e-01  9.979294e-02  1.234646e-02
 [61]  1.205822e-01 -2.795662e-03 -8.579814e-02 -4.381426e-02 -1.073630e-02
 [66]  1.081820e-02  2.666187e-02 -9.525477e-02 -1.584691e-01 -1.127760e-01
 [71] -5.629630e-02  5.907902e-02 -5.795741e-02  9.204867e-03  1.468181e-02
 [76] -1.403657e-01  3.622606e-02  6.612426e-03 -4.633993e-03  1.349325e-02
 [81] -3.932505e-02 -4.939290e-02 -3.537326e-02 -3.268520e-02 -5.724633e-02
 [86] -1.432599e-01  5.820650e-02 -2.223518e-02  2.126983e-02  1.638898e-02
 [91] -9.992723e-02 -1.292766e-01 -6.136741e-02  9.720248e-02 -5.135433e-02
 [96] -5.707568e-02 -1.671088e-02 -9.247370e-02 -2.865466e-02 -1.282820e-01
[101] -4.647897e-02 -7.348502e-02 -1.645871e-01 -1.503995e-01  9.181503e-03
[106] -5.303451e-02  8.576450e-02  1.610014e-01  5.641788e-02  4.915538e-01
[111] -6.240783e-02  5.341302e-02  1.272695e-01 -7.841060e-02 -1.750503e-01
[116] -1.287805e-01  8.270677e-02  4.633570e-01 -8.006866e-02  4.242125e-02
# Generate the hist graph and density line
hist(AAPLmr,col=8,ylim=c(0,6),prob=TRUE)
lines(density(AAPLmr),col="red",lwd=3)
```



Fit an appropriate distribution
Because there are negative numbers, a normal distribution may be the best choice, let's see whether it is.

```
library(fitdistrplus)
fitAAPLmrn<-fitdist(AAPLmr,"norm")
plot(fitAAPLmrn)
par(mfrow=c(2,2))
plot.legend<-c("normal")
cdfcomp(list(fitAAPLmrn),legendtext=c("norm"))
denscomp(list(fitAAPLmrn),legendtext=c("norm"))
qqcomp(list(fitAAPLmrn),legendtext=c("norm"))
ppcomp(list(fitAAPLmrn),legendtext=c("norm"))
```



We can see that it does fit the normal distribution well.
To compute $P(r > 10\%)$ by using normal distribution, we have to calculate the mean and standard deviation and then, standardize it to fit N(0,1).

```
mean(AAPLmr)
[1] -0.01102323
var(AAPLmr)
[1] 0.009584318
sd(AAPLmr)
[1] 0.09789953
```

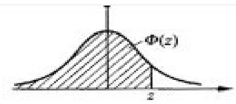
AAPLmr$\sim$N(-0.011009,0.009584268)

$$\phi(0.1) = P\left(x \leq 0.1\right) = P\left(\frac{x + 0.011009}{0.09789927} \leq \frac{0.1 + 0.011009}{0.09789927} = 1.13391\right)$$

$\phi(1.13391) \in (\phi(1.1), \phi(1.2))$ $P(r \leq 10\%) \in (0.8643 \sim 0.8849)$

$P(r > 10\%) \in (0.1151 \sim 0.1357)$

$$\Phi(z)=\int_{-\infty}^{z}\frac{1}{\sqrt{2\pi}}e^{-\frac{u^2}{2}}du=P\{Z\leqslant z\}$$



z	0	1	2	3	4	5	6	7	8	9
0. 0	0. 5000	0. 5040	0. 5080	0. 5120	0. 5160	0. 5199	0. 5239	0. 5279	0. 5319	0. 5359
0. 1	0. 5398	0. 5438	0. 5478	0. 5517	0. 5557	0. 5596	0. 5636	0. 5675	0. 5714	0. 5753
0. 2	0. 5793	0. 5832	0. 5871	0. 5910	0. 5948	0. 5987	0. 6026	0. 6064	0. 6103	0. 6141
0. 3	0. 6179	0. 6217	0. 6255	0. 6293	0. 6331	0. 6368	0. 6406	0. 6443	0. 6480	0. 6517
0. 4	0. 6554	0. 6591	0. 6628	0. 6664	0. 6700	0. 6736	0. 6772	0. 6808	0. 6844	0. 6879
0. 5	0. 6915	0. 6950	0. 6985	0. 7019	0. 7054	0. 7088	0. 7123	0. 7157	0. 7190	0. 7224
0. 6	0. 7257	0. 7291	0. 7324	0. 7357	0. 7389	0. 7422	0. 7454	0. 7486	0. 7517	0. 7549
0. 7	0. 7580	0. 7611	0. 7642	0. 7673	0. 7703	0. 7734	0. 7764	0. 7794	0. 7823	0. 7852
0. 8	0. 7881	0. 7910	0. 7939	0. 7967	0. 7995	0. 8023	0. 8051	0. 8078	0. 8106	0. 8133
0. 9	0. 8159	0. 8186	0. 8212	0. 8238	0. 8264	0. 8289	0. 8315	0. 8340	0. 8365	0. 8389
1. 0	0. 8413	0. 8438	0. 8461	0. 8485	0. 8508	0. 8531	0. 8554	0. 8577	0. 8599	0. 8621

续表

z	0	1	2	3	4	5	6	7	8	9
1. 1	0. 8643	0. 8665	0. 8686	0. 8708	0. 8729	0. 8749	0. 8770	0. 8790	0. 8810	0. 8830
1. 2	0. 8849	0. 8869	0. 8888	0. 8907	0. 8925	0. 8944	0. 8962	0. 8980	0. 8997	0. 9015
1. 3	0. 9032	0. 9049	0. 9066	0. 9082	0. 9099	0. 9115	0. 9131	0. 9147	0. 9162	0. 9177
1. 4	0. 9192	0. 9207	0. 9222	0. 9236	0. 9251	0. 9265	0. 9278	0. 9292	0. 9306	0. 9319
1. 5	0. 9332	0. 9345	0. 9357	0. 9370	0. 9382	0. 9394	0. 9406	0. 9418	0. 9430	0. 9441
1. 6	0. 9452	0. 9463	0. 9474	0. 9484	0. 9495	0. 9505	0. 9515	0. 9525	0. 9535	0. 9545
1. 7	0. 9554	0. 9564	0. 9573	0. 9582	0. 9591	0. 9599	0. 9608	0. 9616	0. 9625	0. 9633
1. 8	0. 9641	0. 9648	0. 9656	0. 9664	0. 9671	0. 9678	0. 9686	0. 9693	0. 9700	0. 9706
1. 9	0. 9713	0. 9719	0. 9726	0. 9732	0. 9738	0. 9744	0. 9750	0. 9756	0. 9762	0. 9767
2. 0	0. 9772	0. 9778	0. 9783	0. 9788	0. 9793	0. 9798	0. 9803	0. 9808	0. 9812	0. 9817
2. 1	0. 9821	0. 9826	0. 9830	0. 9834	0. 9838	0. 9842	0. 9846	0. 9850	0. 9854	0. 9857
2. 2	0. 9861	0. 9864	0. 9868	0. 9871	0. 9874	0. 9878	0. 9881	0. 9884	0. 9887	0. 9890
2. 3	0. 9893	0. 9896	0. 9898	0. 9901	0. 9904	0. 9906	0. 9909	0. 9911	0. 9913	0. 9916
2. 4	0. 9918	0. 9920	0. 9922	0. 9925	0. 9927	0. 9929	0. 9931	0. 9932	0. 9934	0. 9936
2. 5	0. 9938	0. 9940	0. 9941	0. 9943	0. 9945	0. 9946	0. 9948	0. 9949	0. 9951	0. 9952
2. 6	0. 9953	0. 9955	0. 9956	0. 9957	0. 9959	0. 9960	0. 9961	0. 9962	0. 9963	0. 9964
2. 7	0. 9965	0. 9966	0. 9967	0. 9968	0. 9969	0. 9970	0. 9971	0. 9972	0. 9973	0. 9974
2. 8	0. 9974	0. 9975	0. 9976	0. 9977	0. 9977	0. 9978	0. 9979	0. 9979	0. 9980	0. 9981
2. 9	0. 9981	0. 9982	0. 9982	0. 9983	0. 9984	0. 9984	0. 9985	0. 9985	0. 9986	0. 9986
3. 0	0. 9987	0. 9990	0. 9993	0. 9995	0. 9997	0. 9998	0. 9998	0. 9999	0. 9999	1. 0000

We can also use R to compute the probability:

```
1-pnorm(0.1,mean=mean(AAPLmr),sd=sd(AAPLmr))
[1] 0.1283862
```

4 R Exercise

To get $P(0.20 \leq X \leq 0.50)$, we should compute the value of:

$$\int_{0.2}^{0.5} f(x)dx$$

```
f<-function(x){3*(x^2)}
integrate(f,lower=0.2,upper=0.5)
0.117 with absolute error < 1.3e-15
```

The numerical answer is 0.117 up to a small error 1.310^{-15}.

5 R Exercise

a. Find c
To get c analytically, I use the Substitution Technique of Polar Coordinate.
Let $x = \cos\theta, y = r\sin\theta, (0 \le \theta \le 2\pi, 0 \le r \le 1)$
$dxdy = rdrd\theta$, and so that $x^2 + y^2 \le 1$

$\int\int f(x,y)dxdy = 1$

c$\times\int_0^{2\pi}\int_0^1 \sqrt{1-r^2}rdrd\theta = 1$

$= c \times \int_0^{2\pi} d\theta \int_0^1 \sqrt{1-r^2}rdr$

$= c \times \int_0^{2\pi} \left[-\frac{1}{3}(1-r^2)^{\frac{3}{2}}|_0^1\right] d\theta$

$= c \times \int_0^{2\pi} \frac{1}{3}d\theta$

$= c \times \frac{1}{3} \times \theta|_0^{2\pi} = \frac{2}{3}c\pi$

So $c \times \dfrac{2\pi}{3} = 1$, $c = \dfrac{3}{2\pi}$

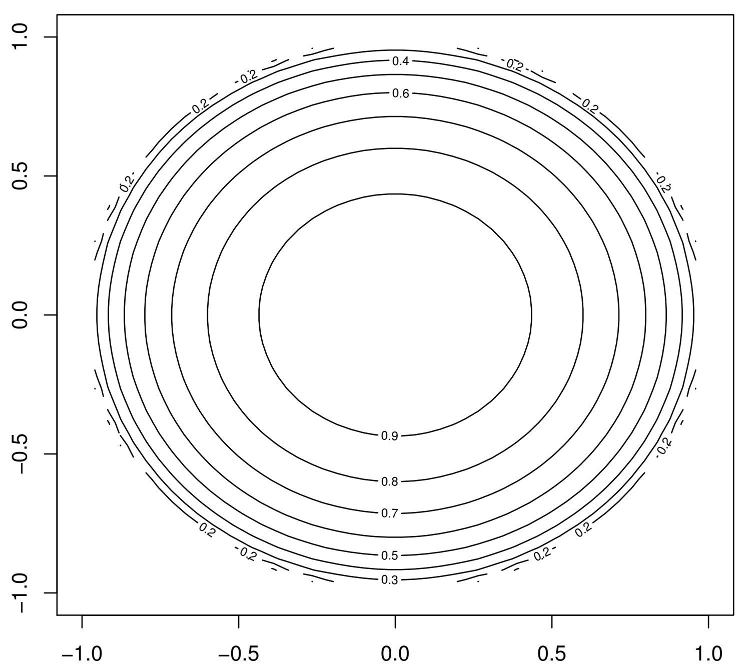
To calculate it numerically in R, there are two possible ways:

```
# define the function
f0<-function(x,y) sqrt(1-x^2-y^2)
# let x^2+Y^2<=1
fp<-function(x,y) y*f0(y*cos(x), y*sin(x))
# compute the double integral
integrate(function(y) {
sapply(y, function(y) {
integrate(function(x) fp(x,y),0,2*pi)$value
})
},0,1)
2.094395 with absolute error < 0.00025
# Or use the function "quad2d"
install.packages("pracma")
library(pracma)
f0<-function(x,y) sqrt(1-x^2-y^2)
fp<-function(x,y) y*f0(y*cos(x),y*sin(x))
quad2d(fp,0,2*pi,0,1)
[1] 2.094422
```

So $c \approx \dfrac{1}{2.0944} \approx \dfrac{3}{2\pi}$

b. Sketch the joint density function
Also, there are two possible ways:

```
# https://stat.ethz.ch/pipermail/r-help/2008-September/173858.html
x<-seq(-1,1,len=50)
y<-seq(-1,1,len=50)
z<-outer(x,y,function(x, y){
sqrt(1-x^2-y^2)
})
contour(x,y,z,xlim=c(-1,1),ylim=c(-1,1))
# https://www.rdocumentation.org/packages/LaplacesDemon/versions
/16.0.1/topics/joint.density.plot
f0<-function(x,y) sqrt(1-x^2-y^2)
fp<-function(x,y) y*f0(y*cos(x), y*sin(x))
joint.density.plot(x,y,Title="Joint Density Plot",contour=TRUE)
```



c. Find $P(X^2 + Y^2) \leq \frac{1}{2}$

Same as a, we can change the upper limit of r to $\sqrt{\frac{1}{2}}$.

$$\frac{3}{2\pi} \times \int_0^{2\pi} \int_0^{\sqrt{\frac{1}{2}}} \sqrt{1 - r^2} r dr d\theta = \frac{4 - \sqrt{2}}{4} \approx 0.6464466$$

Using R:

```
f0<-function(x,y) sqrt(1-x^2-y^2)
fp<-function(x,y) y*f0(y*cos(x),y*sin(x))
quad2d(fp,0,2*pi,0,sqrt(2)/2)*(3/(2*pi))
[1] 0.6464466
```

d. Find the marginal densities of X and Y

because $x^2 + y^2 \leq 1$, so $-\sqrt{1-x^2} \leq y \leq \sqrt{1-x^2}$

$f_X(x) = (x, y)dy$

$= \int_{-\sqrt{1-x^2}}^{\sqrt{1-x^2}} \frac{3}{2\pi} \times \sqrt{1 - x^2 - y^2} dy$

$= \frac{3}{2\pi} \left[\frac{1}{2} y - \sqrt{1 - x^2 - y^2} + \frac{1}{2}(1 - x^2) \arcsin \frac{y}{\sqrt{1-x^2}} \right] |_{-\sqrt{1-x^2}}^{\sqrt{1-x^2}}$

$= \frac{3}{2\pi} \times \frac{1}{2}(1 - x^2)\pi = \frac{3}{4}(1 - x^2)$

$f_Y(y) = \frac{3}{4}(1 - y^2)$, (Based on symmetrical characteristic)

$f_X(x) f_Y(y) = \dfrac{3}{4}(1 - x^2) \times \dfrac{3}{4}(1 - x^2) = \dfrac{9}{16}(1 - x^2)(1 - y^2) \neq f(x, y)$

So they are not independent.

e. Find the conditional densities.

$$f_{X|Y}(x|y) = \frac{f(x, y)}{f_Y(y)} = \frac{\frac{3}{2\pi}\sqrt{1 - x^2 - y^2}}{\frac{3}{4}(1 - y^2)} = 2\pi \sqrt{1 - x^2 - y^2}(1 - y^2)$$

$$f_{Y|X}(y|x) = \frac{f(x, y)}{f_X(x)} = \frac{\frac{3}{2\pi}\sqrt{1 - x^2 - y^2}}{\frac{3}{4}(1 - x^2)} = 2\pi \sqrt{1 - x^2 - y^2}(1 - x^2)$$

6 The expected value of fortune after n rounds of plays

1st Round:

$$E_1 = \frac{1}{2} \times 2c + \frac{1}{2} \times \frac{1}{2}c = \frac{5}{4}c$$

2nd Round:

$$E_2 = \frac{1}{2} \times \frac{5}{4}c \times 2 + \frac{1}{2} \times \frac{5}{4}c \times \frac{1}{2} = \frac{25}{16}c$$

2nd Round:

$$E_3 = \frac{1}{2} \times \frac{25}{16}c \times 2 + \frac{1}{2} \times \frac{25}{16}c \times \frac{1}{2} = \frac{125}{64}c$$

......

Actually we can see that it is a geometric progression with a common ratio of $\frac{5}{4}$.

So After n rounds, the expectation will be:

$$E_n = \left(\frac{5}{4}\right)^n \times c$$

7 X and Y have a bivariate normal density

a. Find $E(X|Y)$

$$X \sim N(0, \sigma^2), Y \sim N(0, \tau^2)$$

$$f_{X|Y}(x|y) = \frac{f(x,y)}{f_Y(y)}$$

$$= \frac{\frac{1}{2\pi\sigma\tau\sqrt{1-\rho^2}} e^{-\frac{1}{2(1-\rho^2)}\left[\frac{x^2}{\sigma^2} - 2\rho\frac{xy}{\sigma\tau} + \frac{y^2}{\tau^2}\right]}}{\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{y^2}{2\tau^2}}}$$

$$= \frac{1}{2\sqrt{2\pi}\sigma\sqrt{1-\rho^2}} e^{-\frac{1}{2(1-\rho^2)}\left(\frac{x}{\sigma} - \rho\frac{y}{\tau}\right)^2}$$

$$= \frac{1}{2\sqrt{2\pi}\sigma\sqrt{1-\rho^2}} e^{-\frac{1}{2\sigma^2(1-\rho^2)}\left(x - \frac{\sigma}{\tau}\rho y\right)^2}$$

So, when Y=y, $x \sim N\left(\frac{\sigma}{\tau}\rho y, \sigma^2(1-\rho^2)\right)$
$E(X|Y=y) = \frac{\sigma}{\tau}\rho y$
$E(X|Y) = \frac{\sigma}{\tau}\rho Y$

b. Find $var(X|Y)$

Based on the result we get from a., the value of $var(X|Y)$ is $\sigma^2(1-\rho^2)$.

c. Find $E(X|X+Y=z)$

Let $X = aU + bV$ and $X + Y = cU$, $U \sim N(0,1)$ and $Y \sim N(0,1)$. U and V are independent. We can get the following equations:

$var(X) = E(x^2) - (Ex)^2 = E(a^2U^2 + 2abUV + b^2V^2) - 0 = a^2 \times 1 + b^2 \times 1 = a^2 + b^2$

$cov(X, X+Y) = EX(X+Y) - EXEY = E(X^2 + XY) - 0 = \sigma^2 + \rho\sigma\tau$

$cov(aU + bV, cU) = E(aU + bV)cU - E(aU + bV)E(cU) = ac \times 1 + 0 - 0 = ac$

$var(X+Y) = D(X) + D(Y) + 2cov(X,Y) = \sigma^2 + \tau^2 + 2\rho$

$var(cU) = c^2 \times 1 = c^2$

We can get:

$a^2 + b^2 = \sigma^2,$

$\sigma^2 + \rho\sigma\tau = ac,$

$\sigma^2 + \tau^2 + 2\rho = c^2,$

so that

$a = \frac{\sigma^2 + \rho\sigma\tau}{\sqrt{\sigma^2 + 2\rho\sigma\tau + \tau^2}}, b = \frac{\sqrt{1-\rho^2}\sigma\tau}{\sqrt{\sigma^2 + 2\rho\sigma\tau + \tau^2}}, c = \sqrt{\sigma^2 + 2\rho\sigma\tau + \tau^2}$
$E(X+Y|X+Y=z) = E\left(aU + bV|U = \frac{z}{c}\right)$
$= E\left(aU|U = \frac{z}{c}\right) + E\left(bV|U = \frac{z}{c}\right) = E\left(aU|U = \frac{z}{c}\right) + 0$
$= \frac{a}{c}z$
$= \frac{\sigma^2 + \rho\sigma\tau}{\sigma^2 + 2\rho\sigma\tau + \tau^2}z$

d. Find $var(X|X+Y=z)$

$$var(X|X+Y=z) = E(X^2|X+Y=z) - [E(X|X+Y=z)]^2$$

$$= E(a^2U^2 + 2abUV + b^2V^2|U = \frac{z}{c} - \left(\frac{a}{z}c\right)^2$$

$$= \frac{a^2z^2}{c^2} + b^2 - \left(\frac{a}{z}c\right)^2$$

$$= \frac{(1-\rho^2)\sigma^2\tau^2}{\sigma^2 + 2\rho\sigma\tau + \tau^2}$$

8 Solve this problem in R

The R code is:

```
# get a set of data fitting normal distribution
X<-rnorm(10000)
Y<-rnorm(10000)
# pick up the ordinal number of a qualified number
n<-which((X+Y)>1-0.0001&(X+Y)<1+0.0001)
E<-mean((X[n])^2)
E
[1] 0.197253
```

The result may change every time.