Monthly_Average_Salary_Analysis

Wharton Chan

17 February 2016

Pre-Processing

First load required libraries:

```r
library(ggplot2)
library(RColorBrewer)
library(data.table)
library(reshape2)
```

Load in data, which has been pre-processed in Excel to give a .csv

```r
setwd("~/Gary/Data_Analysis")
data_wage = read.csv("monthly_average_salary.csv",header = T, as.is = T,na.strings = "N.A.")
```

Create column for time objects for time series plotting:

```r
date_cols = data_wage[,1:2]
date_cols$conc_date = paste(date_cols$Year,date_cols$Month,"01",sep = " ")
date_cols$date = as.Date(date_cols$conc_date,"%Y %b %d")
data_wage = cbind(date_cols$date,data_wage[,3:length(data_wage)])
colnames(data_wage)[1] = "Date"
```

From the raw data the occupations are also subdivided into groups. Create such list:

```r
occ_list = list()
occ_list[["sup_tech"]] = c("office_supervisor", "account_supervisor",
    "estate_officer")
occ_list[["clerical"]] = c("stock_clerk", "account_clerk", "general_clerk",
    "cashier_clerk", "receptionist", "shipping_clerk", "restaurant_receptionist")
occ_list[["service"]] = c("cook", "waiter", "security_guard")
occ_list[["misc"]] = c("general_worker", "office_assistant",
    "driver", "dishwasher", "lav_cleaner", "gen_cleaner")
```

Account for Inflation

Load in data for CPI, and create time objects similarly:

```r
data_cpi = read.csv("CPI_monthly_1990_2015.csv",,header = T, as.is = T,na.strings = "N.A.")
date_cols = data_cpi[,1:2]
date_cols$conc_date = paste(date_cols$Year,date_cols$Month,"01",sep = " ")
date_cols$date = as.Date(date_cols$conc_date,"%Y %b %d")
data_cpi = cbind(date_cols$date,data_cpi[,3:length(data_cpi)])
colnames(data_cpi)[1] = "Date"
```

To calculate real prices, we align the prices to the latest CPI, which will be the 20150901 timepoint. It has been decided that basket A will be used (temporarily) given the low income nature of the list of occupations within the dataset.

```
curr_cpi = data_cpi[which(data_cpi$Date == "2015-09-01"), "A_index"]
data_cpi$A_ratio = curr_cpi/data_cpi$A_index
data_adj = data_cpi[, c("Date", "A_ratio")]
data_wage2 = merge(data_wage, data_adj, by = "Date", all.x = T)
# multiply prices with column A_ratio
data_wage2 = data.table(data_wage2)
data_wage_adjA = data_wage2[, .(office_supervisor = office_supervisor *
    A_ratio, account_supervisor = account_supervisor * A_ratio,
    estate_officer = estate_officer * A_ratio, stock_clerk = stock_clerk *
        A_ratio, account_clerk = account_clerk * A_ratio, general_clerk = general_clerk *
        A_ratio, cashier_clerk = cashier_clerk * A_ratio, receptionist = receptionist *
        A_ratio, shipping_clerk = shipping_clerk * A_ratio, restaurant_receptionist = restaurant_receptionist
        A_ratio, cook = cook * A_ratio, waiter = waiter * A_ratio,
    security_guard = security_guard * A_ratio, general_worker = general_worker *
        A_ratio, office_assistant = office_assistant * A_ratio,
    driver = driver * A_ratio, dishwasher = dishwasher * A_ratio,
    lav_cleaner = lav_cleaner * A_ratio, gen_cleaner = gen_cleaner *
        A_ratio), by = Date]
```

To compare the best CPI to use, we also try composite CPI:

```
curr_cpi = data_cpi[which(data_cpi$Date == "2015-09-01"), "comp_index"]
data_cpi$comp_ratio = curr_cpi/data_cpi$comp_index
data_adj = data_cpi[, c("Date", "comp_ratio")]
data_wage2 = merge(data_wage, data_adj, by = "Date", all.x = T)
# multiply prices with column comp_ratio
data_wage2 = data.table(data_wage2)
data_wage_adjcomp = data_wage2[, .(office_supervisor = office_supervisor *
    comp_ratio, account_supervisor = account_supervisor * comp_ratio,
    estate_officer = estate_officer * comp_ratio, stock_clerk = stock_clerk *
        comp_ratio, account_clerk = account_clerk * comp_ratio,
    general_clerk = general_clerk * comp_ratio, cashier_clerk = cashier_clerk *
        comp_ratio, receptionist = receptionist * comp_ratio,
    shipping_clerk = shipping_clerk * comp_ratio, restaurant_receptionist = restaurant_receptionist *
        comp_ratio, cook = cook * comp_ratio, waiter = waiter *
        comp_ratio, security_guard = security_guard * comp_ratio,
    general_worker = general_worker * comp_ratio, office_assistant = office_assistant *
        comp_ratio, driver = driver * comp_ratio, dishwasher = dishwasher *
        comp_ratio, lav_cleaner = lav_cleaner * comp_ratio, gen_cleaner = gen_cleaner *
        comp_ratio), by = Date]
```

Now change data structure in both data frames to allow efficient plotting:

```
adjcomp = melt(data_wage_adjcomp,id.var = "Date")
adjA = melt(data_wage_adjA,id.var = "Date")
colnames(adjcomp) = c("Date","Occupation","Salary")
colnames(adjA) = c("Date","Occupation","Salary")
```

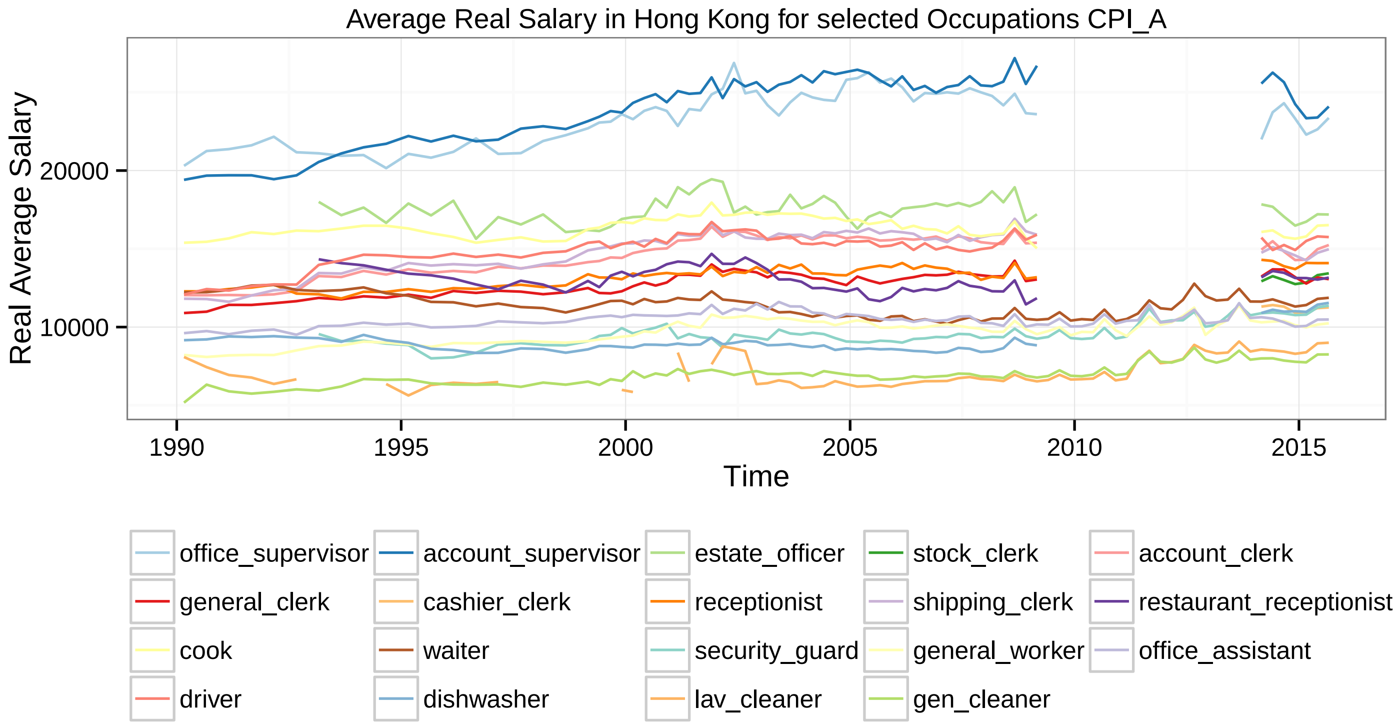
Plots

There are 19 occupations, so first define the colours:

```
set_colour = c(brewer.pal(12,"Paired"),brewer.pal(7,"Set3"))
names(set_colour) = colnames(data_wage_adjA)[2:length(data_wage_adjA)]
```

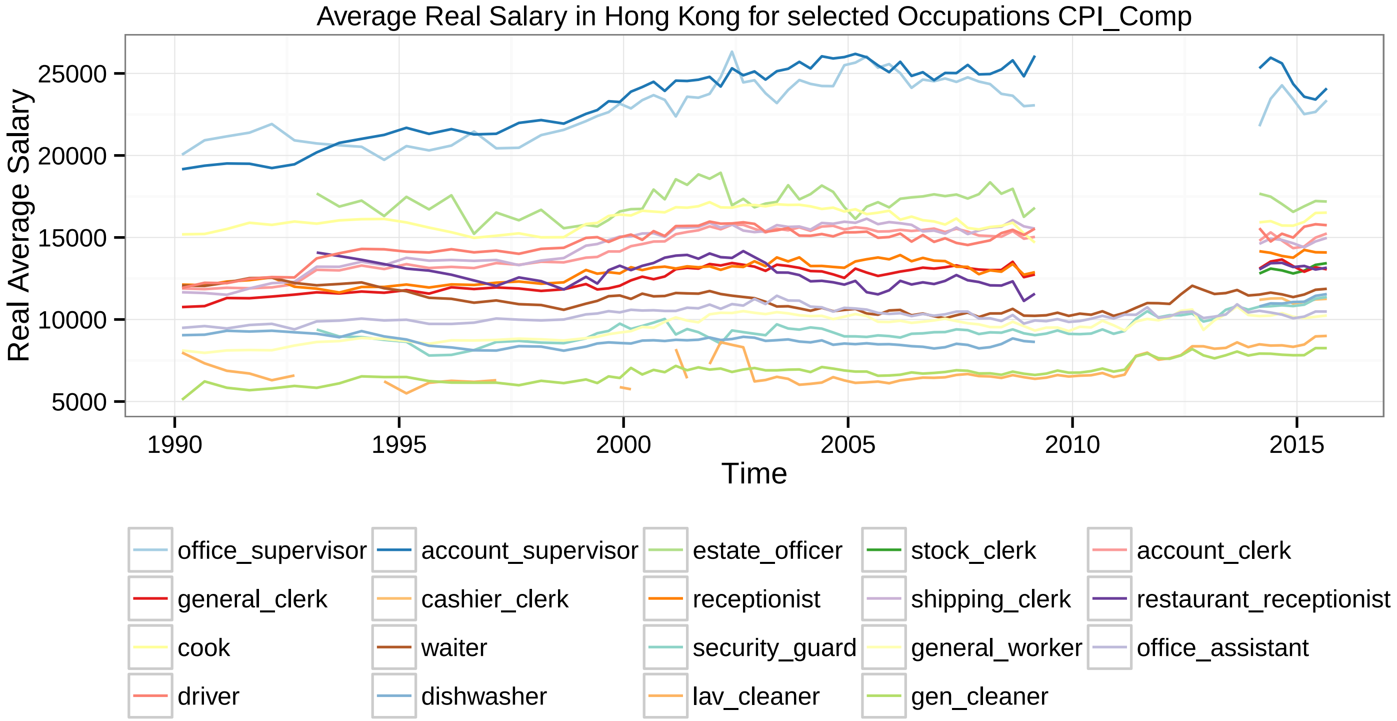
Plot for df adjusted with basket A:

```
ggplot(adjA, aes(x = Date, y = Salary, colour = Occupation)) +
    geom_line() + scale_x_date() + labs(x = "Time", y = "Real Average Salary",
    title = "Average Real Salary in Hong Kong for selected Occupations CPI_A") +
    scale_colour_manual("variable", values = set_colour) + theme_bw() +
    theme(legend.position = "bottom", plot.title = element_text(size = rel(0.9))) +
    guides(colour = guide_legend(nrow = 4, byrow = TRUE, title = NULL))
```



Plot for df adjusted with composite basket:

```
ggplot(adjcomp, aes(x = Date, y = Salary, colour = Occupation)) +
    geom_line() + scale_x_date() + labs(x = "Time", y = "Real Average Salary",
    title = "Average Real Salary in Hong Kong for selected Occupations CPI_Comp") +
    scale_colour_manual("variable", values = set_colour) + theme_bw() +
    theme(legend.position = "bottom", plot.title = element_text(size = rel(0.9))) +
    guides(colour = guide_legend(nrow = 4, byrow = TRUE, title = NULL))
```


Average Real Salary in Hong Kong for selected Occupations CPI_Comp
Real Average Salary
25000
20000
15000
10000
5000
1990
1995
2000
2005
2010
2015
Time
office_supervisor
account_supervisor
estate_officer
stock_clerk
account_clerk
general_clerk
cashier_clerk
receptionist
shipping_clerk
restaurant_receptionist
cook
waiter
security_guard
general_worker
office_assistant
driver
dishwasher
lav_cleaner
gen_cleaner