# Income_Distribution_test

### Load a sample dataset

```
read.csv("Input_Data/Wider_aggregated_deciles.csv", stringsAsFactors = FALSE) %>%
  select(country, year, gdp_ppp_pc_usd2011, gini) %>%
  distinct() %>%
  mutate(sce="Historical data") %>%
  filter(year > 2013)->data_for_lognorm
head(data_for_lognorm)
```

```
##     country year gdp_ppp_pc_usd2011     gini             sce
## 1 Argentina 2014            18.9490 0.396005 Historical data
## 2 Australia 2014            43.5615 0.342125 Historical data
## 3   Austria 2014            43.8475 0.298130 Historical data
## 4   Austria 2015            44.0700 0.297710 Historical data
## 5   Belgium 2014            41.3035 0.274710 Historical data
## 6   Belgium 2015            41.7700 0.271340 Historical data
```

## 1. Generate deciles using lognormal approach

```
##      country year category pred_shares
## 1: Argentina 2014       d1  0.02176029
## 2: Argentina 2014       d2  0.03495252
## 3: Argentina 2014       d3  0.04502104
## 4: Argentina 2014       d4  0.05983864
## 5: Argentina 2014       d5  0.06788144
## 6: Argentina 2014       d6  0.08301407
##                                        model
## 1: Log normal based downscaling (using country GINI)
## 2: Log normal based downscaling (using country GINI)
## 3: Log normal based downscaling (using country GINI)
## 4: Log normal based downscaling (using country GINI)
## 5: Log normal based downscaling (using country GINI)
## 6: Log normal based downscaling (using country GINI)
```

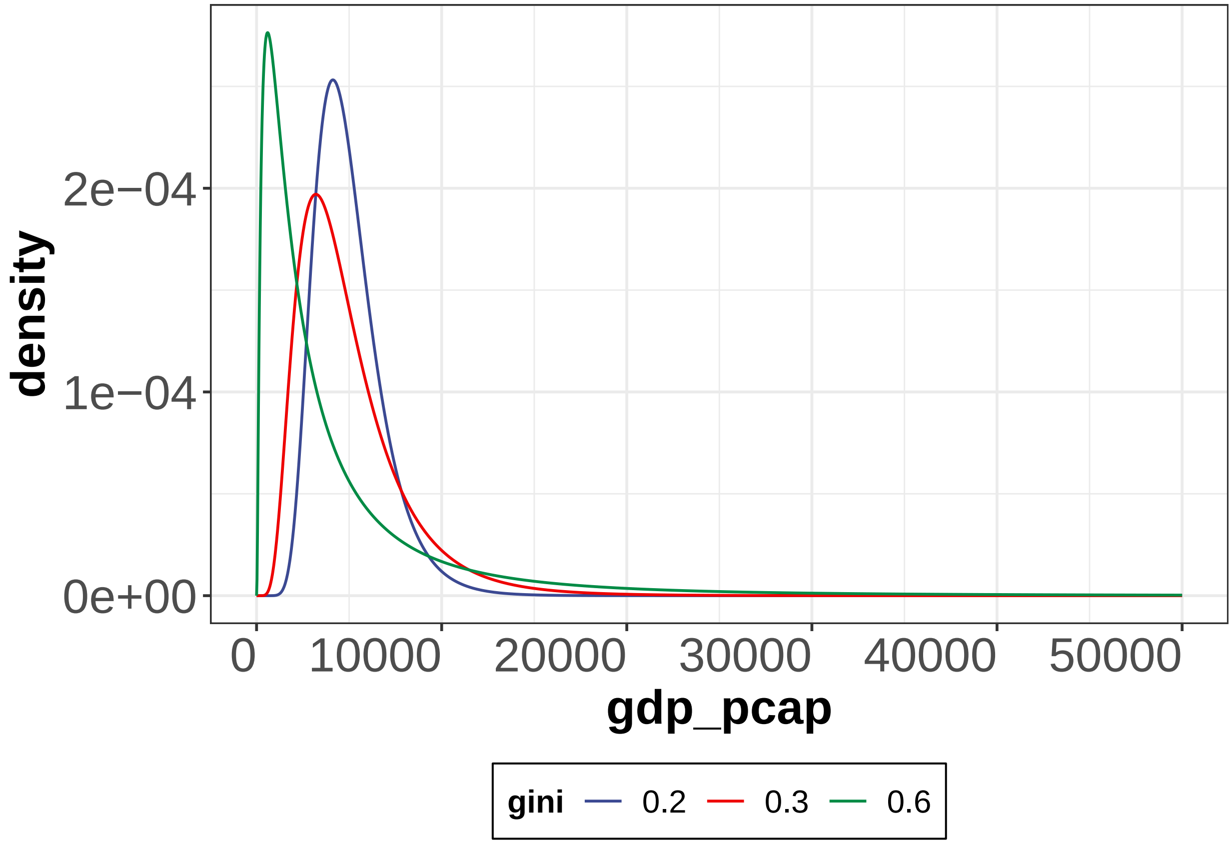
```
## [1] "Processed in 4 seconds"
```

## 2. Generate lognormal density dist

```
density_dist_1 <- compute_lognormal_dist(mean_income = 5000,gini=0.6) %>% mutate(gini=as.character(0.6))
density_dist_2 <- compute_lognormal_dist(mean_income = 5000,gini=0.3)%>% mutate(gini=as.character(0.3))
density_dist_3 <- compute_lognormal_dist(mean_income = 5000,gini=0.2)%>% mutate(gini=as.character(0.2))
```

```r
g <- ggplot(data=bind_rows(density_dist_3,density_dist_2,density_dist_1), aes(x=gdp_pcap,y=density,colo
     geom_line()+scale_color_aaas()

g+scheme_basic
```




**Compile data**

```r
IV_data <- compile_independent_variables() %>% mutate(Component1_act= Component1, Component2_act= Compo
```

```
## [1] "There are negative values for 8 observations. Please use the correction function to correct neg
## [1] "There are still 1 negative observations. Restarting loop."

## Joining, by = c("country", "year")

## [1] "There are negative values for 20 observations. Please use the correction function to correct ne
## [1] "There are still 6 negative observations. Restarting loop."
## [1] "There are still 1 negative observations. Restarting loop."

## Joining, by = c("country", "year")
```

```r
Final_historical_data <- read.csv("Input_Data/Final_Historical_data_ISO.csv", stringsAsFactors = FALSE)
   select(year,gini=output_name,iso,country) %>%
   distinct() %>%
   left_join(IV_data %>% select(-gini,-country), by= c("iso","year")) %>%
   mutate(sce= "Historical data") %>%
   select(country,year,iso,sce,gini,labsh,lagged_ninth_decile,lagged_palma_ratio) %>%
   group_by(iso,sce) %>%
   mutate(labsh = ifelse(is.na(labsh),approx_fun(year,labsh,rule = 2),labsh),
          lagged_ninth_decile = ifelse(is.na(lagged_ninth_decile),approx_fun(year,lagged_ninth_decile,ru
          lagged_palma_ratio=ifelse(is.na(lagged_palma_ratio),approx_fun(year,lagged_palma_ratio,rule=2)
   ungroup() %>%
  group_by(year) %>%
  arrange(gini) %>%
   mutate(labsh = ifelse(is.na(labsh),0,labsh),
          lagged_ninth_decile =ifelse(is.na(lagged_ninth_decile),0,lagged_ninth_decile),
          lagged_palma_ratio=ifelse(is.na(lagged_palma_ratio),0,lagged_palma_ratio)) %>%
  ungroup() %>%
   distinct() %>%
   na.omit()


Final_historical_data %>% filter(iso %in% c("ind","chn","usa"))->sample_data

head(sample_data)
```

```
## # A tibble: 6 x 8
##   country  year iso   sce              gini labsh lagged_ninth_de~ lagged_palma_ra~
##   <chr>   <int> <chr> <chr>           <dbl> <dbl>            <dbl>            <dbl>
## 1 China    1977 chn   Historical data 0.186 0.592            0.147            0.904
## 2 China    1978 chn   Historical data 0.212 0.592            0.147            0.904
## 3 China    1976 chn   Historical data 0.225 0.592            0.147            0.904
## 4 China    1982 chn   Historical data 0.231 0.592            0.147            0.904
## 5 China    1980 chn   Historical data 0.233 0.592            0.147            0.904
## 6 China    1992 chn   Historical data 0.235 0.575            0.148            1.22
```

## 3. Run PC model

```r
start_time <- Sys.time()
PC_model_results <- PC_model(sample_data)
```

```
## [1] "Now generating deciles"
```

```r
head(PC_model_results)
```

```
## # A tibble: 6 x 7
##   country  year sce             Category pred_shares Component1 Component2
##   <chr>   <int> <chr>           <chr>          <dbl>      <dbl>      <dbl>
## 1 India    1968 Historical data d1            0.0136       4.13      -1.21
## 2 India    2015 Historical data d1            0.0155       3.37      -1.03
## 3 India    2014 Historical data d1            0.0156       3.37      -1.07
```

```
## 4 India    2013 Historical data d1           0.0159       3.37      -1.15
## 5 China    2009 Historical data d1           0.0162       3.17      -1.05
## 6 India    2012 Historical data d1           0.0163       3.36      -1.31
```

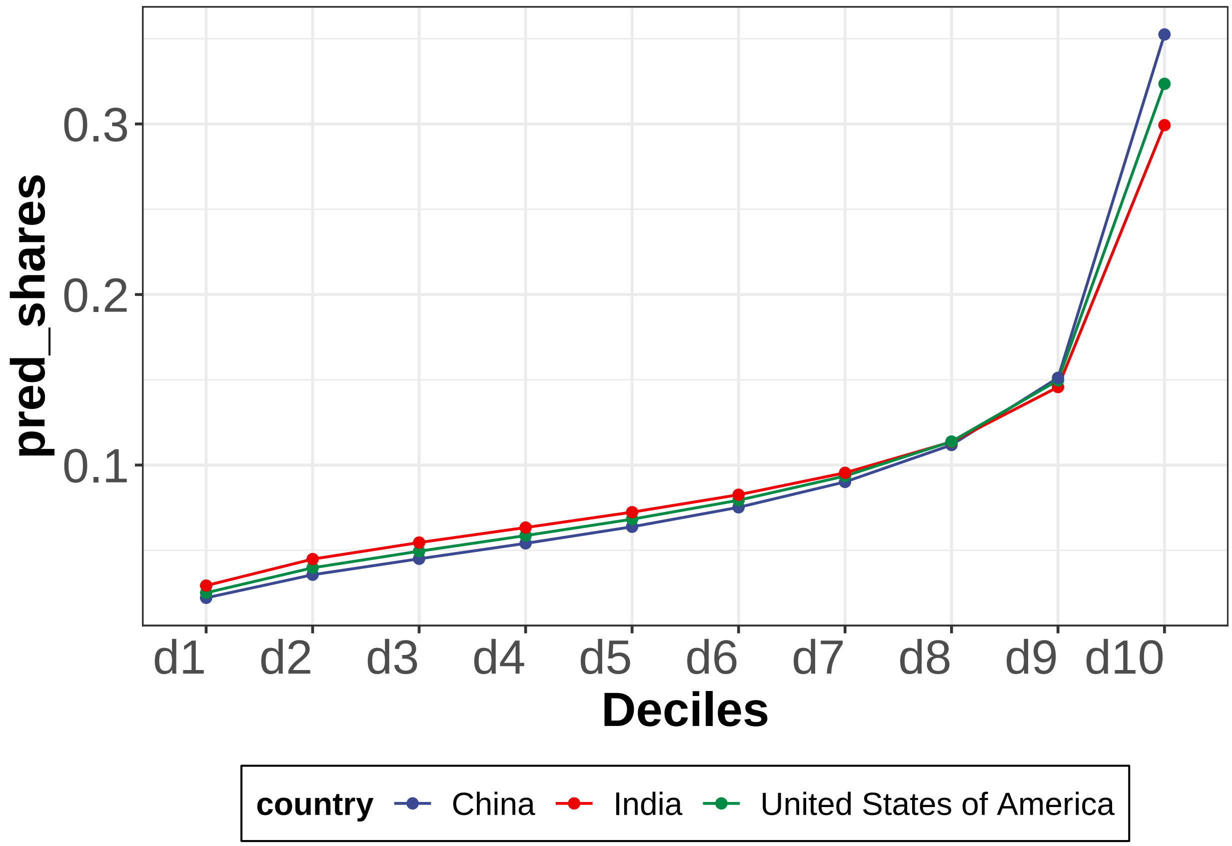
```
print(paste0("Completed in ", as.integer(Sys.time()-start_time), " seconds."))
```

```
## [1] "Completed in 5 seconds."
```

### Plot deciles

```
g <- ggplot(data=PC_model_results %>% filter(year==2010),aes(x=factor(Category,levels =c('d1','d2','d3'
  'd5','d6','d7','d8',                                                                               '
     geom_line()+
     geom_point()+scale_color_aaas()+xlab("Deciles")

g+scheme_basic
```




## 4. Generate GINI for a given set of deciles

```
gini_data <-compute_gini_deciles(PC_model_results %>% mutate(category=Category), inc_col = "pred_shares
head(gini_data)
```

```
## # A tibble: 6 x 11
```

```
##   country  year sce             Category pred_shares Component1 Component2 category
##   <chr>   <int> <chr>           <chr>          <dbl>      <dbl>      <dbl> <chr>
## 1 India    1968 Historical data d1            0.0136       4.13      -1.21 d1
## 2 India    2015 Historical data d1            0.0155       3.37      -1.03 d1
## 3 India    2014 Historical data d1            0.0156       3.37      -1.07 d1
## 4 India    2013 Historical data d1            0.0159       3.37      -1.15 d1
## 5 China    2009 Historical data d1            0.0162       3.17      -1.05 d1
## 6 India    2012 Historical data d1            0.0163       3.36      -1.31 d1
## # ... with 3 more variables: share_of_richer_pop <dbl>, score <dbl>,
## #   output_name <dbl>
```

### 5. Aggregate deciles to a region

```
read.csv("Input_Data/Wider_aggregated_deciles.csv", stringsAsFactors = FALSE) %>%
  select(country, year, gdp_ppp_pc_usd2011, population,Income..net.,Category) %>% filter(year%in% c(2015

head(ISO_data)
```

```
##    country year gdp_ppp_pc_usd2011 population Income..net. Category
## 1  Austria 2015            44.0700   17357324   0.02990000       d1
## 2  Belgium 2015            41.7700   22575871   0.03430000       d1
## 3  Bolivia 2015             6.4880   21594437   0.01110000       d1
## 4   Brazil 2015            14.7535  410433868   0.01250000       d1
## 5    Chile 2015            22.3865   35732037   0.01826667       d1
## 6 Colombia 2015            13.0185   95749365   0.01190000       d1
##   GCAM_region_ID
## 1         Global
## 2         Global
## 3         Global
## 4         Global
## 5         Global
## 6         Global
```

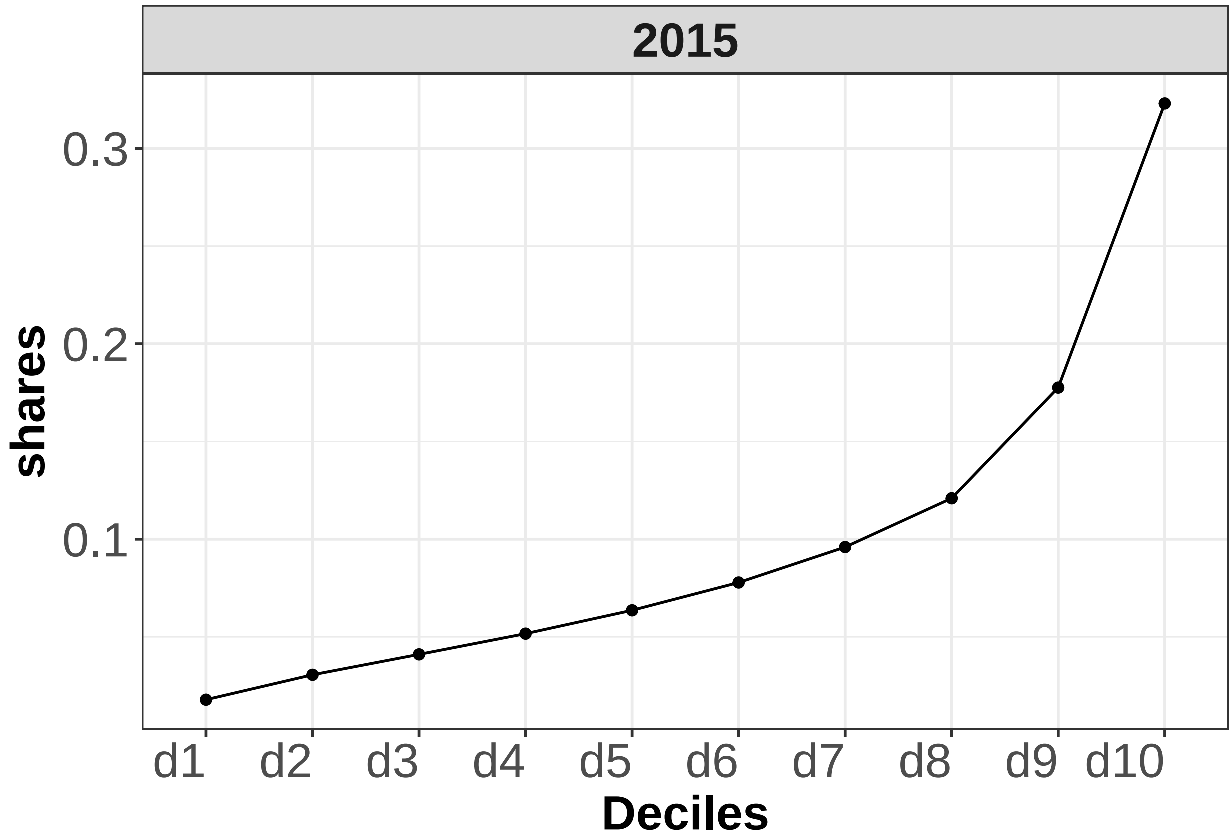
```
aggregate_country_deciles_to_regions(ISO_data,grouping_variables = c("GCAM_region_ID","year"))->agg_data
head(agg_data)
```

```
## # A tibble: 6 x 10
##   GCAM_region_ID  year category shares gdp_pcap       tot_gdp tot_pop share_of_richer~
##   <chr>          <int> <chr>     <dbl>    <dbl>         <dbl>   <dbl>            <dbl>
## 1 Global          2015 d1       0.0178     14.7 176725558262. 1.20e10              0.9
## 2 Global          2015 d2       0.0306     14.7 176725558262. 1.20e10              0.8
## 3 Global          2015 d3       0.0410     14.7 176725558262. 1.20e10              0.7
## 4 Global          2015 d4       0.0516     14.7 176725558262. 1.20e10              0.6
## 5 Global          2015 d5       0.0636     14.7 176725558262. 1.20e10              0.5
## 6 Global          2015 d6       0.0778     14.7 176725558262. 1.20e10              0.4
## # ... with 2 more variables: score <dbl>, output_name <dbl>
```

```
g <- ggplot(data=agg_data,aes(x=factor(category,levels =c('d1','d2','d3','d4',
  'd5','d6','d7','d8',                                                                                     '
    geom_line()+
```

```
    geom_point()+scale_color_aaas()+xlab("Deciles")+facet_wrap(~year)

g+scheme_basic
```




### 6. Model inter-operability

```
library(ambrosia)

PC_model_results %>%
  filter(country=="India",year==2010) %>%
  mutate(gpp_pc = (5000*1000000*pred_shares)/100000,
         Ps=0.6,
         Pn=1.2)->data_for_food_model

head(data_for_food_model)
```

```
## # A tibble: 6 x 10
##   country  year sce       Category pred_shares Component1 Component2 gpp_pc    Ps
##   <chr>   <int> <chr>     <chr>          <dbl>      <dbl>      <dbl>  <dbl> <dbl>
## 1 India    2010 Histori~ d1            0.0293     -0.817      -1.28  1467.   0.6
## 2 India    2010 Histori~ d2            0.0449     -0.817      -1.28  2243.   0.6
## 3 India    2010 Histori~ d3            0.0546     -0.817      -1.28  2728.   0.6
## 4 India    2010 Histori~ d4            0.0634     -0.817      -1.28  3169.   0.6
```

```
## 5 India    2010 Histori~ d5           0.0724     -0.817      -1.28  3619.   0.6
## 6 India    2010 Histori~ d6           0.0826     -0.817      -1.28  4129.   0.6
## # ... with 1 more variable: Pn <dbl>
```

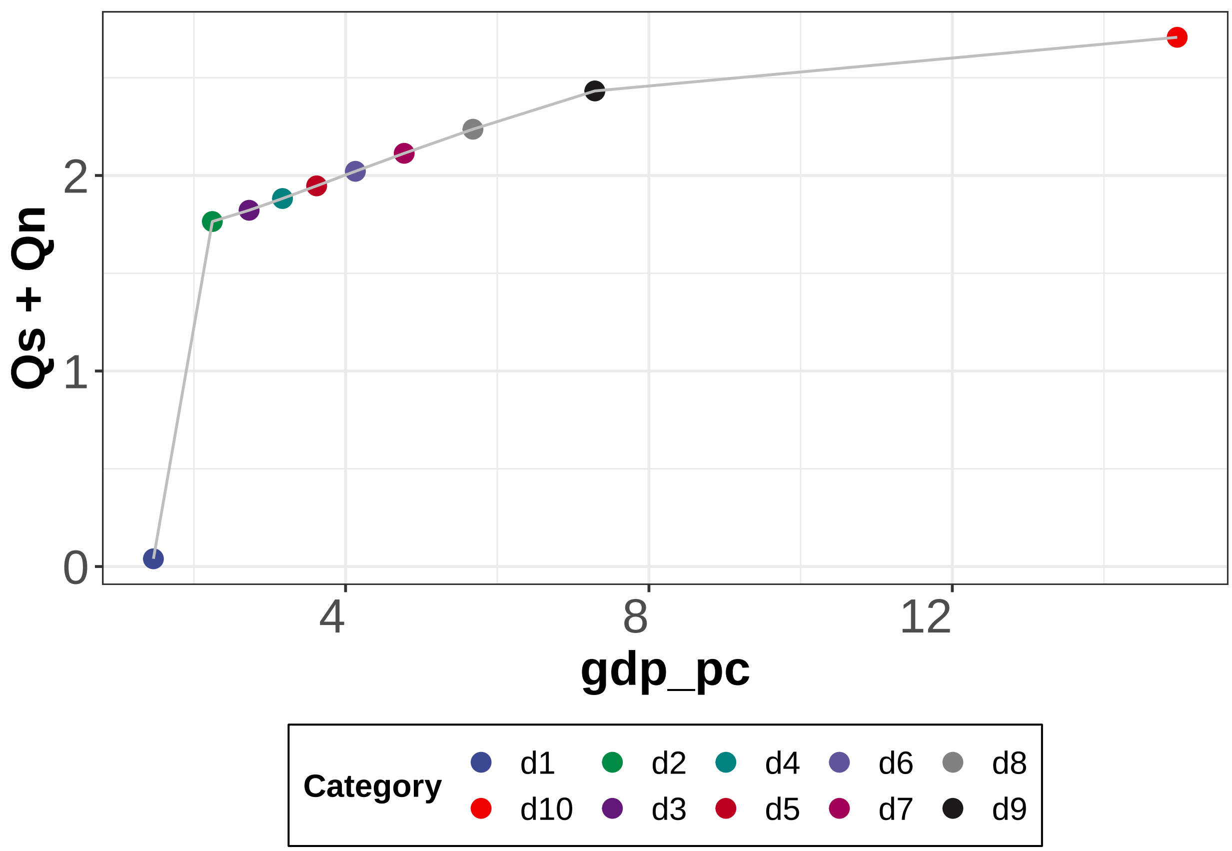
```
food_demand <- food.dmnd(data_for_food_model$Ps,data_for_food_model$Pn,data_for_food_model$gpp_pc/1000)
head(food_demand)
```

```
##           Qs        Qn          Qm     alpha.s     alpha.n   alpha.m
## 1 0.01439844 0.0251110 0.05576553 0.005890658 0.02054671 0.9735626
## 2 1.43144419 0.3331681 0.03845696 0.382857715 0.17821996 0.4389223
## 3 1.38900928 0.4331264 0.05370814 0.305469334 0.19050534 0.5040253
## 4 1.36570121 0.5164778 0.06754335 0.258610870 0.19560176 0.5457874
## 5 1.35188414 0.5950722 0.08176075 0.224156703 0.19733855 0.5785047
## 6 1.34415331 0.6772149 0.09804468 0.195302675 0.19679583 0.6079015
```

```
food_demand$gdp_pc <- data_for_food_model$gpp_pc/1000
food_demand$Category <- data_for_food_model$Category
```

```
g <- ggplot(data=food_demand %>% mutate(id="A"), aes(x=gdp_pc,y=Qs+Qn,color=Category,group=id))+
     geom_point(size=3)+scale_color_aaas()+geom_line(color="grey")

g+scheme_basic
```



### 7. Generate results using single component

```r
get_deciles_from_components(PC_model_results,use_second_comp = FALSE,grouping_variables = c("country","y

head(Single_comp_results)
```

```
## # A tibble: 6 x 6
##   country  year Category pred_shares Component1 Component2
##   <chr>   <int> <chr>          <dbl>      <dbl>      <dbl>
## 1 India    1968 d1            0.0103       4.13      -1.21
## 2 India    2015 d1            0.0127       3.37      -1.03
## 3 India    2014 d1            0.0127       3.37      -1.07
## 4 India    2013 d1            0.0127       3.37      -1.15
## 5 China    2009 d1            0.0133       3.17      -1.05
## 6 India    2012 d1            0.0127       3.36      -1.31
```