Midterm 2

Hector Carvajal

2025-02-28

Data

For this midterm you need to use two datasets:

"chinalanduse_MODIS_2012.nc" contains four layers with land cover data for China. The data were derived from MODIS satellite data for the year 2012. Each layer gives the fraction of the grid cell that has a specific land cover type: urban (layer 1), cropland (layer 2), grassland (layer 3) and forest (layer 4).

"ch_adm.*" with polygons for the provinces of China.

Q1. Read in the land use data as a SpatRaster get the polygons as a SpatVector (2 points)

```r
getwd()
```

```
## [1] "/Users/hectorsmacbookpro/Documents/School/ESP 106/Midterms/Midterm 2/midterm2esp106"
```

```r
list.files()
```

```
## [1] "chinalanduse_MODIS_2012.nc" "chn_adm.cpg"
## [3] "chn_adm.dbf"                "chn_adm.prj"
## [5] "chn_adm.shp"                "chn_adm.shx"
## [7] "Midterm 2.Rmd"              "Midterm2ESP106.pdf"
## [9] "Midterm2ESP106.Rmd"
```

```r
library(terra)
```

```
## terra 1.8.21
```

```r
land_use <- rast("/Users/hectorsmacbookpro/Documents/School/ESP 106/Midterms/Midterm 2/midterm2esp106/c
provinces <- vect("/Users/hectorsmacbookpro/Documents/School/ESP 106/Midterms/Midterm 2/midterm2esp106/
print(land_use)
```

```
## class       : SpatRaster
## dimensions  : 1860, 14001, 4  (nrow, ncol, nlyr)
## resolution  : 0.0256, 0.0208  (x, y)
## extent      : -178.4628, 179.9628, 14.9993, 53.6873  (xmin, xmax, ymin, ymax)
## coord. ref. : lon/lat WGS 84 (CRS84) (OGC:CRS84)
## source      : chinalanduse_MODIS_2012.nc
## varname     : variable
## names       : variable_z=1, variable_z=2, variable_z=3, variable_z=4
```

```r
print(provinces)
```

```
##  class       : SpatVector
##  geometry    : polygons
##  dimensions  : 33, 11  (geometries, attributes)
##  extent      : 73.5577, 134.7739, 18.15931, 53.56086  (xmin, xmax, ymin, ymax)
##  source      : chn_adm.shp
##  coord. ref. : lon/lat WGS 84 (EPSG:4326)
##  names       :    NAME_1   GID_1 GID_0 COUNTRY VARNAME_1 NL_NAME_1    TYPE_1
##  type        :     <chr>   <chr> <chr>   <chr>     <chr>     <chr>     <chr>
##  values      :     Anhui CHN.1_1   CHN   China     Ānhuī   |      Shěng
##                  Beijing CHN.2_1   CHN   China   Běijīng   |  Zhíxiáshì
##                Chongqing CHN.3_1   CHN   China Chóngqìng   |  Zhíxiáshì
##     ENGTYPE_1  CC_1 HASC_1 ISO_1
##         <chr> <chr>  <chr> <chr>
##      Province    NA  CN.AH CN-AH
##  Municipality    NA  CN.BJ CN-BJ
##  Municipality    NA  CN.CQ CN-CQ
```

Q2a. Crop the land use SpatRaster to the same extent as the SpatVector of Chinese provinces (1 point), and set all grid cells outside of China to `NA`

```r
#crops the raster to the extent of the SpatVector of chinese provinces
land_use_cropped <- crop(land_use, provinces)



#masks the raster using the provinces SpatVector and sets the cells outide of china
land_use_masked <- mask(land_use_cropped, provinces)
print(land_use_masked)
```

```
## class       : SpatRaster
## dimensions  : 1702, 2391, 4  (nrow, ncol, nlyr)
## resolution  : 0.0256, 0.0208  (x, y)
## extent      : 73.56923, 134.7788, 18.1609, 53.5625  (xmin, xmax, ymin, ymax)
## coord. ref. : lon/lat WGS 84 (CRS84) (OGC:CRS84)
## source(s)   : memory
## varname     : variable
## names       : variable_z=1, variable_z=2, variable_z=3, variable_z=4
## min values  :            0,            0,            0,            0
## max values  :            1,            1,            1,            1
```

Q2b. Rename the layers in the SpatRaster so they provide information about what data is in each of the 4 layers (2 points)

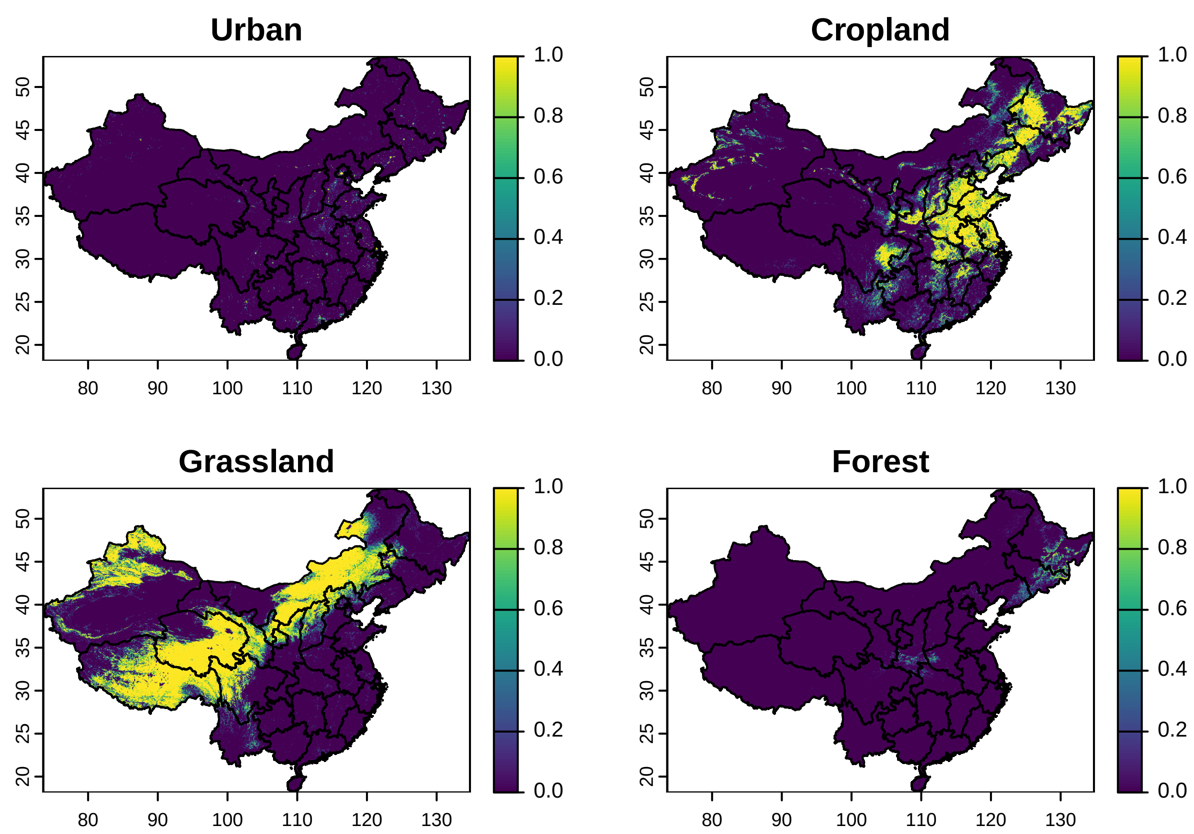
```r
#renames the layers of the SpatRaster
names(land_use_masked) <- c("Urban", "Cropland", "Grassland", "Forest")
print(land_use_masked)
```

```
## class       : SpatRaster
## dimensions  : 1702, 2391, 4  (nrow, ncol, nlyr)
```

```
## resolution  : 0.0256, 0.0208  (x, y)
## extent      : 73.56923, 134.7788, 18.1609, 53.5625  (xmin, xmax, ymin, ymax)
## coord. ref. : lon/lat WGS 84 (CRS84) (OGC:CRS84)
## source(s)   : memory
## varname     : variable
## names       : Urban, Cropland, Grassland, Forest
## min values  :     0,        0,         0,      0
## max values  :     1,        1,         1,      1
```

Q3. Make a figure showing each SpatRaster layer on one of the panels and overlay the polygons of the Chinese provinces. Title each panel with the type of land use it shows. (4 points)

```r
#rname the layers to proper land use names
names(land_use_masked) <- c("Urban", "Cropland", "Grassland", "Forest")
#2x2 plotting layout
par(mfrow = c(2, 2))
#layer names
layer_names <- names(land_use_masked)

#loops and plots
for (i in 1:4) {
  plot(land_use_masked[[i]], main = layer_names[i])
#overlays province borders
  plot(provinces, add = TRUE, border = "black", lwd = 1)
}
```



```r
#searched up this solution online since it kept crashing.
par(mfrow = c(1, 1))
```

Q4a. Use `terra::extract` to find, for each province, the fraction of land in these four classes. [For this question you can assume all the grid cells have the same size] (3 points)

```r
#used terra::extract to get raster values for each province
land_use_values <- extract(land_use_masked, provinces, fun = mean, na.rm = TRUE)
#adds the province names to the extracted data
land_use_values$Province <- provinces$NAME_1
print(land_use_values)
```

```
##    ID        Urban    Cropland   Grassland       Forest        Province
## 1   1 1.485168e-02 0.646405733 0.003847900 1.387219e-03           Anhui
## 2   2 1.153714e-01 0.299729895 0.144125230 3.093923e-03         Beijing
## 3   3 6.521529e-03 0.203184739 0.001481789 5.244315e-03       Chongqing
## 4   4 2.600561e-02 0.061434232 0.004364550 3.092684e-05          Fujian
## 5   5 3.996819e-03 0.087680624 0.400894396 3.527299e-03           Gansu
## 6   6 4.223879e-02 0.119822469 0.009257352 1.260333e-04       Guangdong
## 7   7 1.071244e-02 0.051608185 0.002281014 6.499771e-05         Guangxi
## 8   8 3.764910e-03 0.153545678 0.007175318 3.731592e-04         Guizhou
## 9   9 7.707157e-03 0.086235658 0.002913859 4.523766e-03          Hainan
## 10 10 3.362038e-02 0.483198482 0.306454951 7.317166e-03           Hebei
## 11 11 7.828745e-03 0.389852809 0.033184066 7.253456e-02    Heilongjiang
## 12 12 3.850683e-02 0.782530859 0.009605914 2.720180e-02           Henan
## 13 13 0.000000e+00 0.000000000 0.000000000 0.000000e+00       Hong Kong
## 14 14 1.131149e-02 0.390618615 0.004631285 1.463583e-02           Hubei
## 15 15 9.341245e-03 0.185297511 0.003591011 1.076864e-04           Hunan
## 16 16 3.326057e-02 0.768018677 0.014393223 2.106749e-04         Jiangsu
## 17 17 7.923291e-03 0.222333333 0.005902371 5.718270e-05         Jiangxi
## 18 18 1.181648e-02 0.385392242 0.097113314 1.270092e-01           Jilin
## 19 19 2.529342e-02 0.519516122 0.051588569 6.504468e-02        Liaoning
## 20 20 0.000000e+00 0.033333334 0.026666666 0.000000e+00           Macau
## 21 21 2.629228e-03 0.081413187 0.525555413 1.214391e-03      Nei Mongol
## 22 22 7.515666e-03 0.150782292 0.682385283 1.455427e-04     Ningxia Hui
## 23 23 1.097182e-03 0.002437022 0.689000417 1.850525e-05         Qinghai
## 24 24 8.537541e-03 0.244829928 0.273652750 4.651758e-02         Shaanxi
## 25 25 4.267022e-02 0.851962730 0.028954643 1.324187e-04        Shandong
## 26 26 2.146548e-01 0.553956132 0.024272948 1.949634e-04        Shanghai
## 27 27 2.012471e-02 0.360992363 0.402905064 2.882390e-03          Shanxi
## 28 28 3.773638e-03 0.154516600 0.386690523 1.759347e-03         Sichuan
## 29 29 8.374076e-02 0.669960851 0.080365376 2.435842e-04         Tianjin
## 30 30 1.696892e-03 0.039337617 0.247525979 1.484126e-05 Xinjiang Uygur
## 31 31 6.523447e-05 0.002027606 0.617138677 1.069450e-04          Xizang
## 32 32 7.839736e-03 0.085950516 0.065110655 1.404948e-04          Yunnan
## 33 33 3.024358e-02 0.193322905 0.007327374 2.703911e-04        Zhejiang
```

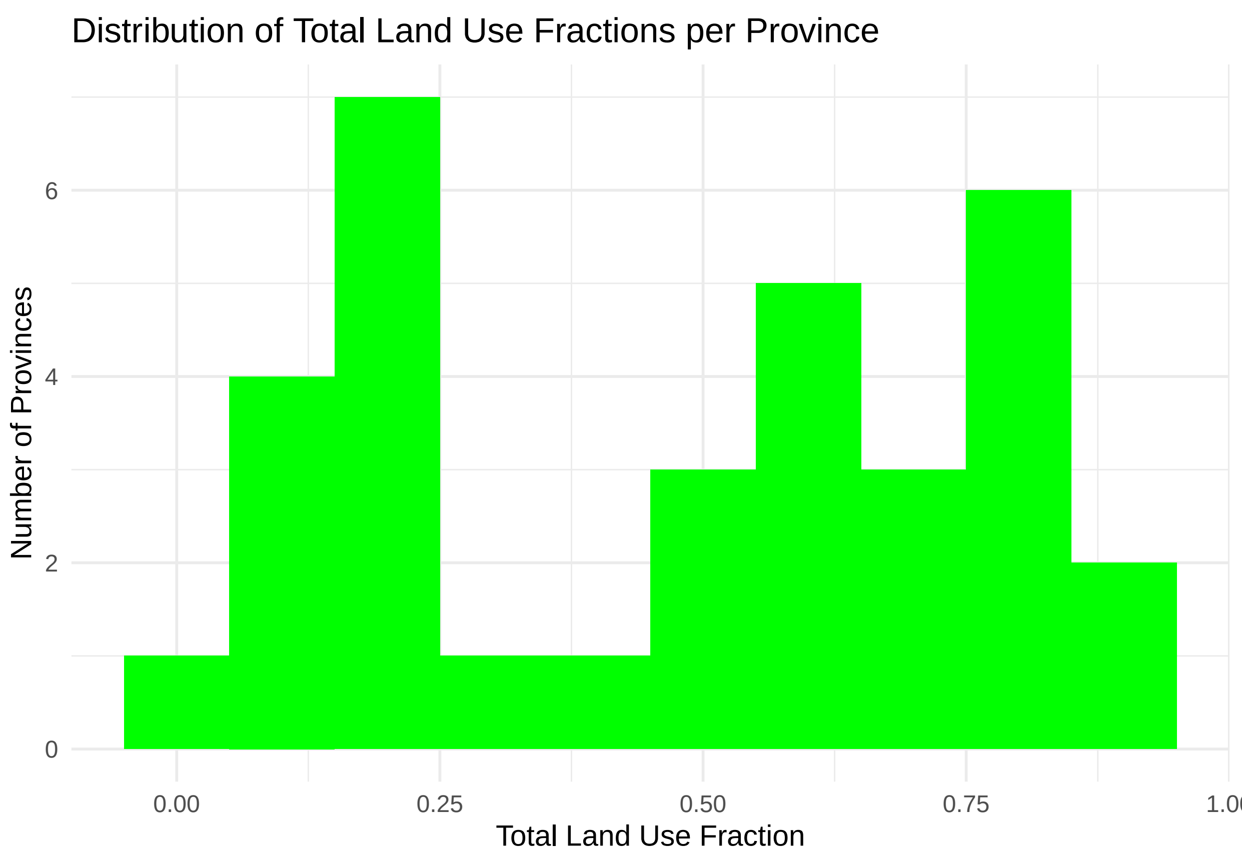
Q4b. Describe the potential problem with the area assumption made in 4a. How might it affect the calculation in that step? What could we do if we didn't want to make that assumption? (You don't have to do it, just describe in theory) (2 points)

Answer: A problem that comes from this assuming that all grid cells are the same size is that their actual area could be different and vary. This means that some provinces might have incorrect land use proportions because grid cells at different latitudes don't represent the same amount of land.

The way i see this working out better is to calculate the actual area of each grid cell using what was given in the instructions in 4a, terra::cellSize() and weigh the values correctly.

Q4c. Sum up the fractions in the four land cover classes for each province and plot these as a histogram. (2 points)

```
library(ggplot2)

#sums the fractions for each province
land_use_values$Total_Fraction <- rowSums(land_use_values[, 2:5], na.rm = TRUE)
#histogram
ggplot(land_use_values, aes(x = Total_Fraction)) +
  geom_histogram(binwidth = 0.1, fill = "green") +
  theme_minimal() +
  labs(title = "Distribution of Total Land Use Fractions per Province",
       x = "Total Land Use Fraction",
       y = "Number of Provinces")
```



Q5. Add a new variable called "other" to the data.frame created with terra::extract. This variable should represent the fraction of all other land cover classes. Assign it the appropriate values. (2 points)

```
#other land cover fraction
land_use_values$Other <- 1 - rowSums(land_use_values[, 2:5], na.rm = TRUE)
print(land_use_values)
```

```
##    ID         Urban    Cropland   Grassland        Forest         Province
```

```
## 1   1 1.485168e-02 0.646405733 0.003847900 1.387219e-03          Anhui
## 2   2 1.153714e-01 0.299729895 0.144125230 3.093923e-03        Beijing
## 3   3 6.521529e-03 0.203184739 0.001481789 5.244315e-03      Chongqing
## 4   4 2.600561e-02 0.061434232 0.004364550 3.092684e-05         Fujian
## 5   5 3.996819e-03 0.087680624 0.400894396 3.527299e-03          Gansu
## 6   6 4.223879e-02 0.119822469 0.009257352 1.260333e-04      Guangdong
## 7   7 1.071244e-02 0.051608185 0.002281014 6.499771e-05        Guangxi
## 8   8 3.764910e-03 0.153545678 0.007175318 3.731592e-04        Guizhou
## 9   9 7.707157e-03 0.086235658 0.002913859 4.523766e-03         Hainan
## 10 10 3.362038e-02 0.483198482 0.306454951 7.317166e-03          Hebei
## 11 11 7.828745e-03 0.389852809 0.033184066 7.253456e-02   Heilongjiang
## 12 12 3.850683e-02 0.782530859 0.009605914 2.720180e-02          Henan
## 13 13 0.000000e+00 0.000000000 0.000000000 0.000000e+00      Hong Kong
## 14 14 1.131149e-02 0.390618615 0.004631285 1.463583e-02          Hubei
## 15 15 9.341245e-03 0.185297511 0.003591011 1.076864e-04          Hunan
## 16 16 3.326057e-02 0.768018677 0.014393223 2.106749e-04        Jiangsu
## 17 17 7.923291e-03 0.222333333 0.005902371 5.718270e-05        Jiangxi
## 18 18 1.181648e-02 0.385392242 0.097113314 1.270092e-01          Jilin
## 19 19 2.529342e-02 0.519516122 0.051588569 6.504468e-02       Liaoning
## 20 20 0.000000e+00 0.033333334 0.026666666 0.000000e+00          Macau
## 21 21 2.629228e-03 0.081413187 0.525555413 1.214391e-03     Nei Mongol
## 22 22 7.515666e-03 0.150782292 0.682385283 1.455427e-04    Ningxia Hui
## 23 23 1.097182e-03 0.002437022 0.689000417 1.850525e-05        Qinghai
## 24 24 8.537541e-03 0.244829928 0.273652750 4.651758e-02        Shaanxi
## 25 25 4.267022e-02 0.851962730 0.028954643 1.324187e-04       Shandong
## 26 26 2.146548e-01 0.553956132 0.024272948 1.949634e-04       Shanghai
## 27 27 2.012471e-02 0.360992363 0.402905064 2.882390e-03         Shanxi
## 28 28 3.773638e-03 0.154516600 0.386690523 1.759347e-03        Sichuan
## 29 29 8.374076e-02 0.669960851 0.080365376 2.435842e-04        Tianjin
## 30 30 1.696892e-03 0.039337617 0.247525979 1.484126e-05 Xinjiang Uygur
## 31 31 6.523447e-05 0.002027606 0.617138677 1.069450e-04         Xizang
## 32 32 7.839736e-03 0.085950516 0.065110655 1.404948e-04         Yunnan
## 33 33 3.024358e-02 0.193322905 0.007327374 2.703911e-04       Zhejiang
##    Total_Fraction      Other
## 1      0.66649253 0.33350747
## 2      0.56232044 0.43767956
## 3      0.21643237 0.78356763
## 4      0.09183531 0.90816469
## 5      0.49609914 0.50390086
## 6      0.17144464 0.82855536
## 7      0.06466663 0.93533337
## 8      0.16485907 0.83514093
## 9      0.10138044 0.89861956
## 10     0.83059097 0.16940903
## 11     0.50340018 0.49659982
## 12     0.85784540 0.14215460
## 13     0.00000000 1.00000000
## 14     0.42119722 0.57880278
## 15     0.19833745 0.80166255
## 16     0.81588315 0.18411685
## 17     0.23621618 0.76378382
## 18     0.62133120 0.37866880
## 19     0.66144279 0.33855721
## 20     0.06000000 0.94000000
```

```
## 21      0.61081222 0.38918778
## 22      0.84082878 0.15917122
## 23      0.69255313 0.30744687
## 24      0.57353780 0.42646220
## 25      0.92372001 0.07627999
## 26      0.79307880 0.20692120
## 27      0.78690453 0.21309547
## 28      0.54674011 0.45325989
## 29      0.83431057 0.16568943
## 30      0.28857533 0.71142467
## 31      0.61933846 0.38066154
## 32      0.15904140 0.84095860
## 33      0.23116425 0.76883575
```

Q6. Make barplots showing the breakdown of urban, cropland, grassland, forest, and other for each province. The barplots should be "stacked" (a single bar for each province, showing land cover with a color) and "horizontal" (province names on the vertical axis).

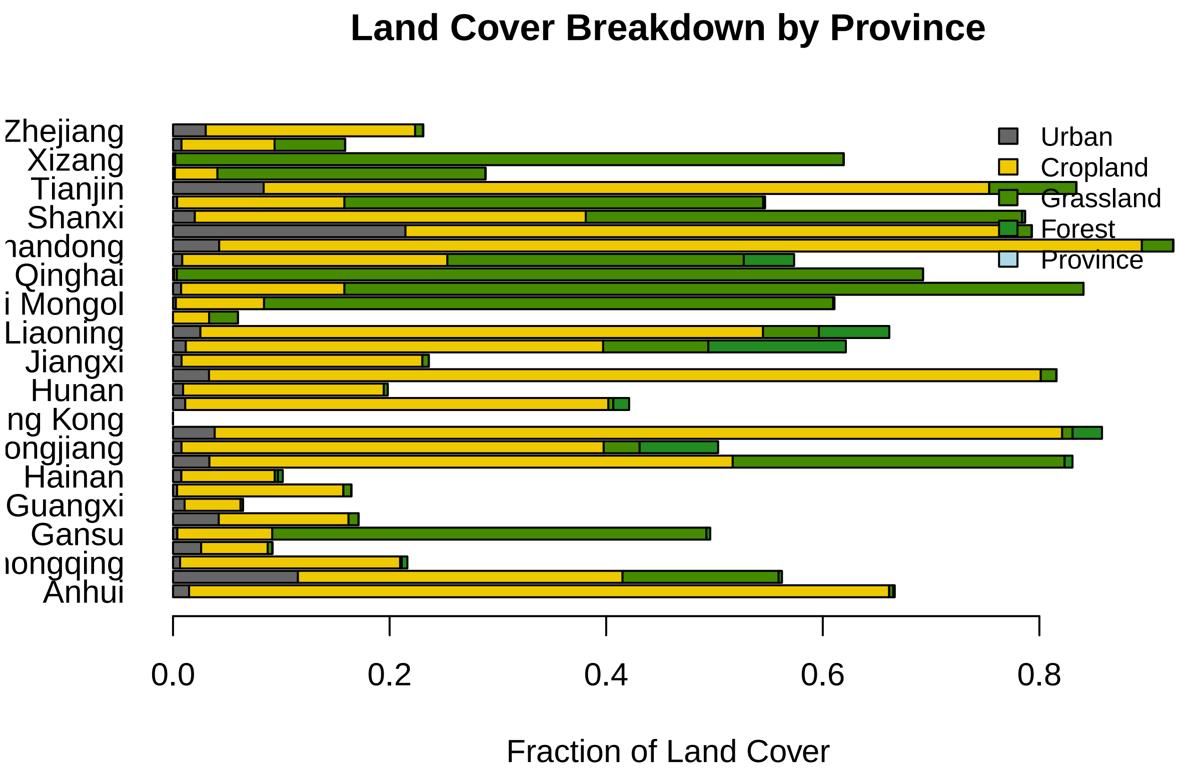
Q6a) Use graphics::barplot, make sure to include a legend. (4 points)

```r
# creates matrix for barplot
land_use_matrix <- as.matrix(land_use_values[, 2:6])
#province names as row labels
province_names <- land_use_values$Province



#rhe different colors for land use types
land_use_colors <- c("gray40", "gold2", "chartreuse4", "forestgreen", "lightblue")

#stacked horizontal barplot
barplot(
  t(land_use_matrix),
  beside = FALSE,
  horiz = TRUE,


  names.arg = province_names,
  col = land_use_colors,

  las = 1,
  xlab = "Fraction of Land Cover",
  main = "Land Cover Breakdown by Province"
)
```

```
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
```

```
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
## Warning in apply(height, 2L, cumsum): NAs introduced by coercion
```

```r
#legemd
legend("topright", legend = colnames(land_use_matrix), fill = land_use_colors, cex = 0.8, bty = "n")
```



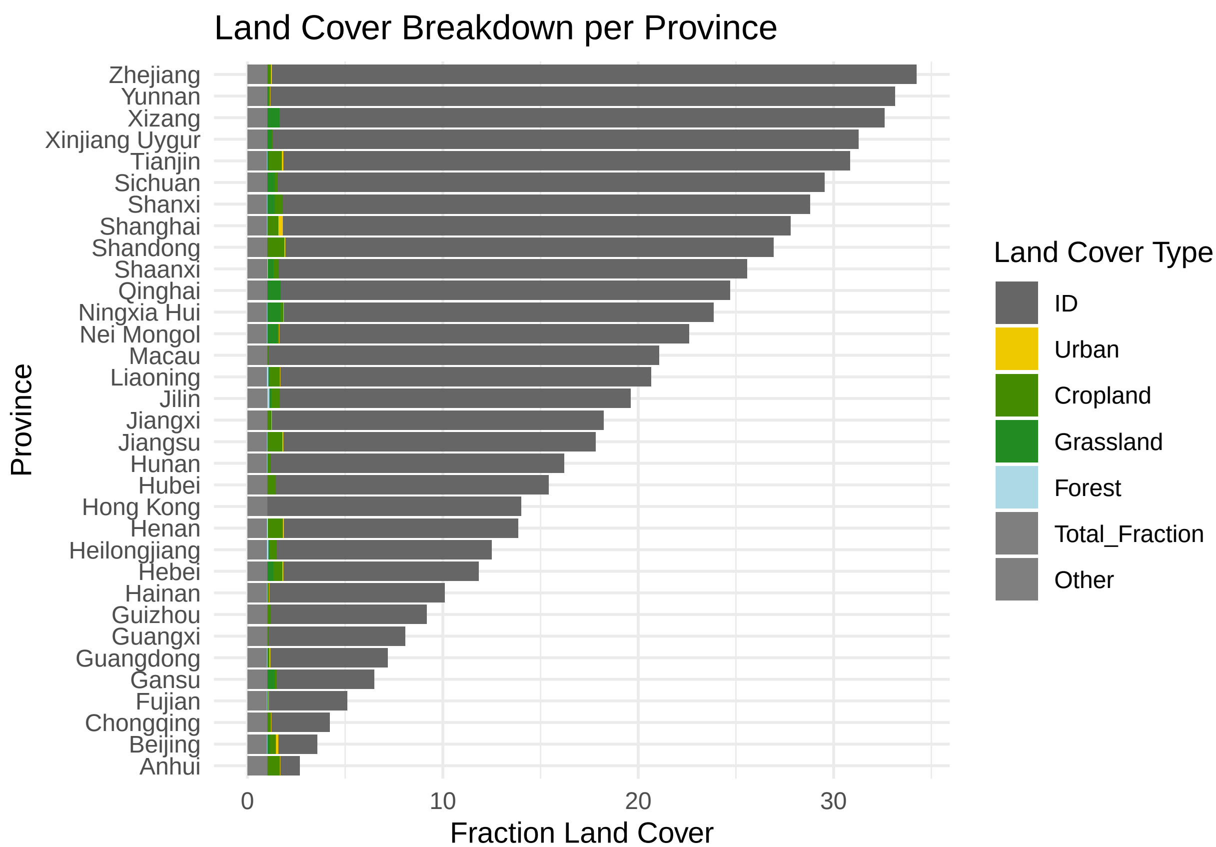
Q6b) Use ggplot. (4 points)

```
library(reshape2)
# reshapes the data to long format for ggplot
land_use_long <- melt(land_use_values, id.vars = "Province",
                      variable.name = "Land_Cover", value.name = "Fraction")


#checks unique land cover types to ensure correct color mapping
unique_land_cover <- unique(land_use_long$Land_Cover)
print(unique_land_cover)
```

```
## [1] ID             Urban          Cropland       Grassland      Forest
## [6] Total_Fraction Other
## Levels: ID Urban Cropland Grassland Forest Total_Fraction Other
```

```
#creates stacked horizontal barploty
ggplot(land_use_long, aes(x = Fraction, y = reorder(Province, Fraction), fill = Land_Cover)) +
  geom_bar(stat = "identity", position = "stack") +

  scale_fill_manual(values = c("gray40", "gold2", "chartreuse4", "forestgreen", "lightblue")[1:length(un


  theme_minimal() +

  labs(title = "Land Cover Breakdown per Province",
```

```
      x = "Fraction Land Cover",
      y = "Province",
      fill = "Land Cover Type")
```



Q7. Upload your R markdown file, and your knitted output to Canvas. Push the R markdown file to your Github repository. (2 points)