

NCEE

Stephanie Yang jy2777

1/16/2019

```
setwd("~/NCEE")
library(ggplot2)
library(gridExtra)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:gridExtra':
##
##      combine

## The following objects are masked from 'package:stats':
##
##      filter, lag

## The following objects are masked from 'package:base':
##
##      intersect, setdiff, setequal, union

library(wesanderson)
library(maps)
library(mapdata)
library(sp)

NCEE_reg <- read.csv("NCEE_reg.csv")[1:31, ]
province <- c(as.character(NCEE_reg[, 1]))
rownames(NCEE_reg) <- province
NCEE_reg <- NCEE_reg[, -1]

NCEE_JHS <- read.csv("NCEE_JHS.csv")

# =====Plot for National Data Trend===== #
NCEE_national <- read.csv("national data.csv")
national_reg <- NCEE_national$Reg[1:14]
national_JHS <- NCEE_national$JHS[7:20]/30000
year_born <- c(2000:1987)

population <- c(national_JHS, national_reg)
label <- c(rep("JHS", 14), rep("NCEE Reg", 14))
year_birth <- rep(year_born, 2)
plot_df <- data.frame(population, label, year_birth)

p1 <- ggplot(data = plot_df, aes(x = year_birth, y = population, fill = label)) +
  geom_bar(stat = "identity", color = "grey", position = position_dodge()) +
  theme_minimal() + scale_fill_brewer(palette = "Blues") + ggtitle("National Data from 1987 to 2000")
  theme(plot.title = element_text(hjust = 0.5)) + xlab("Year of Birth")

# =====Plot for Beijing Data Trend===== #
```

```

Beijing_JHS <- as.numeric(NCEE_JHS[1, 7:19])/30000 #2012-2000
Beijing_reg <- as.numeric(NCEE_reg[1, 1:13])/10000 # 2018-2006
label_city <- c(rep("JHS", 13), rep("NCEE Reg", 13))
year_born <- c(2000:1988)
year_birth <- rep(year_born, 2)
Beijing_population <- c(Beijing_JHS, Beijing_reg)
plot_beijing <- data.frame(Beijing_population, label_city, year_birth)

p2 <- ggplot(data = plot_beijing, aes(x = year_birth, y = Beijing_population,
  fill = label_city)) + geom_bar(stat = "identity", color = "grey", position = position_dodge()) +
  theme_minimal() + scale_fill_brewer(palette = "Reds") + ggtitle("Beijing Data from 1987 to 2000") +
  theme(plot.title = element_text(hjust = 0.5)) + xlab("Year of Birth")

# =====Plot for Henan Data Trend===== #

Henan_JHS <- as.numeric(NCEE_JHS[16, 7:19])/30000 #2012-2000
Henan_reg <- as.numeric(NCEE_reg[16, 1:13])/10000 # 2018-2006
label_city <- c(rep("JHS", 13), rep("NCEE Reg", 13))
year_born <- c(2000:1988)
year_birth <- rep(year_born, 2)
Henan_population <- c(Henan_JHS, Henan_reg)
plot_Henan <- data.frame(Henan_population, label_city, year_birth)

p3 <- ggplot(data = plot_Henan, aes(x = year_birth, y = Henan_population, fill = label_city)) +
  geom_bar(stat = "identity", color = "grey", position = position_dodge()) +
  theme_minimal() + scale_fill_brewer(palette = "Greens") + ggtitle("Henan Data from 1987 to 2000") +
  theme(plot.title = element_text(hjust = 0.5)) + xlab("Year of Birth")

# =====Plot for Shanxi Data Trend===== #

Xinjiang_JHS <- as.numeric(NCEE_JHS[31, 7:19])/30000 #2012-2000
Xinjiang_reg <- as.numeric(NCEE_reg[31, 1:13])/10000 # 2018-2006
label_city <- c(rep("JHS", 13), rep("NCEE Reg", 13))
year_born <- c(2000:1988)
year_birth <- rep(year_born, 2)
Xinjiang_population <- c(Xinjiang_JHS, Xinjiang_reg)
plot_Xinjiang <- data.frame(Xinjiang_population, label_city, year_birth)

p4 <- ggplot(data = plot_Xinjiang, aes(x = year_birth, y = Xinjiang_population,
  fill = label_city)) + geom_bar(stat = "identity", color = "grey", position = position_dodge()) +
  theme_minimal() + scale_fill_brewer(palette = "OrRd") + ggtitle("Xinjiang Data from 1987 to 2000") +
  theme(plot.title = element_text(hjust = 0.5)) + xlab("Year of Birth")

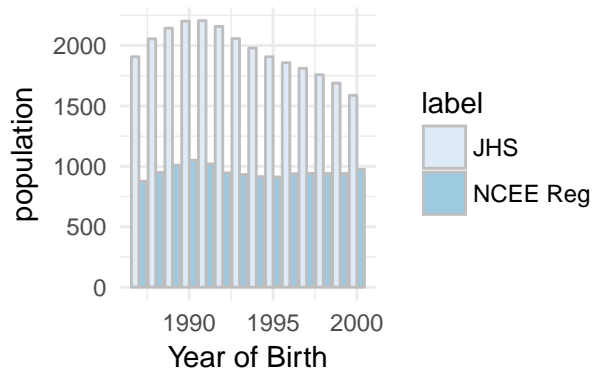
# =====Display===== #
grid.arrange(p1, p2, p3, p4, nrow = 2)

```

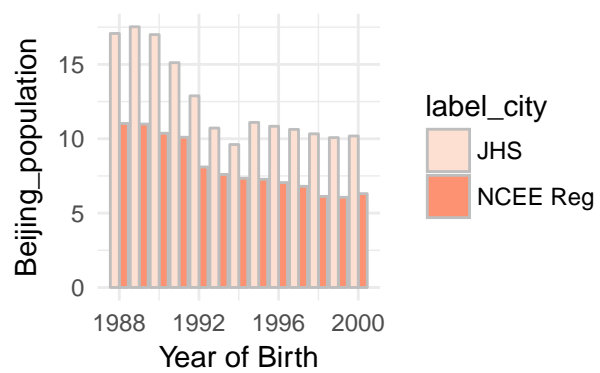
```
## Warning: Removed 1 rows containing missing values (geom_bar).
```

```
## Warning: Removed 1 rows containing missing values (geom_bar).
```

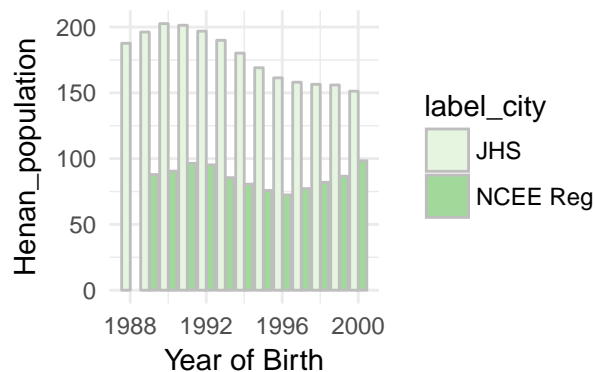
National Data from 1987 to 2000



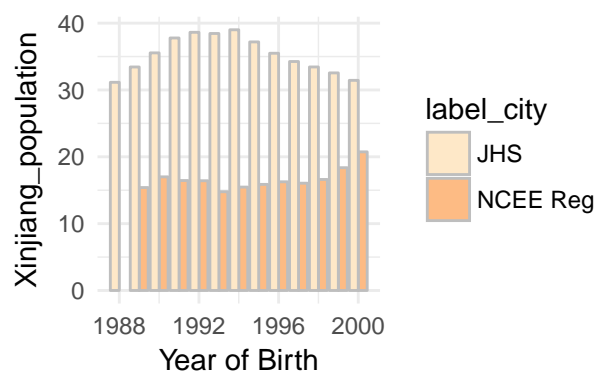
Beijing Data from 1987 to 2000



Henan Data from 1987 to 2000



Xinjiang Data from 1987 to 2000



=====Fit linear model for 31 provinces===== #
NCEE_JHS

##	X	X2017	X2016	X2015	X2014	X2013	X2012	X2011
## 1	Beijing	266404	268273	283366	306789	310568	305510	302269
## 2	Tianjin	262243	256383	261474	267214	260710	256541	261954
## 3	Hebei	2600675	2436810	2361330	2288195	2088470	2173677	2150335
## 4	Shanxi	1082430	1092739	1126849	1218952	1291442	1502433	1643113
## 5	Neimenggu	618655	612376	639648	669657	688464	746308	791411
## 6	Liaoning	963450	978298	1012944	1055661	1057488	1134585	1195997
## 7	Jilin	618704	604277	595518	622883	644993	696588	751532
## 8	Heilongjiang	903983	903883	899803	916293	932839	1204786	1223979
## 9	Shanghai	411712	413298	412345	426789	436696	432686	430585
## 10	Jiangsu	2086934	1949456	1867166	1852029	1857469	1970169	2111249
## 11	Zhejiang	1558460	1504118	1479353	1499062	1482649	1492985	1546002
## 12	Anhui	2021627	1941986	1900786	1924134	1997091	2130347	2498800
## 13	Fujian	1215717	1154758	1133458	1125729	1108226	1120356	1157266
## 14	Jiangxi	1910421	1802080	1763985	1750083	1754361	1945486	2009641
## 15	Shandong	3294601	3159129	3108127	3147954	3179800	3281023	3451577
## 16	Henan	4291617	4158272	4048103	3993606	3850493	4537868	4679780
## 17	Hubei	1487131	1414864	1365319	1375940	1483710	1577701	2040702
## 18	Hunan	2296294	2260603	2224138	2206344	2142847	2111100	2163402
## 19	Guangdong	3561001	3478440	3553170	3767505	4047906	4424650	4790565
## 20	Guangxi	2034632	1987540	1963062	1950844	1950761	1966202	2008317
## 21	Hainan	333342	323736	328889	337350	346790	364677	392397
## 22	Chongqing	990403	966021	960383	979386	1017592	1087258	1190197
## 23	Sichuan	2491364	2448234	2463860	2583315	2717198	3041867	3266108

## 24	Guizhou	1829870	1891411	1979699	2068326	2103033	2100850	2138054
## 25	Yunnan	1872808	1873150	1894282	1897966	1874418	1954348	2052586
## 26	Xizang	124571	120283	117520	124295	126117	130226	136371
## 27	Shaanxi	1049654	1051036	1070779	1117284	1201851	1315464	1498841
## 28	Gansu	856127	876171	909255	970919	1035940	1180171	1285392
## 29	Qinghai	205814	207937	213165	211993	208095	208723	223398
## 30	Ningxia	279180	273696	274333	278323	284758	292813	299635
## 31	Xinjiang	901806	894798	907400	911477	918473	943169	976569
##	X2010	X2009	X2008	X2007	X2006	X2005	X2004	X2003
## 1	309912	318874	325117	332959	288298	321585	386511	453446
## 2	273408	287031	303492	320840	335783	359110	400568	432505
## 3	2212343	2418637	2741801	3062442	3368253	3704263	4035302	4294874
## 4	1713779	1726832	1772798	1851469	1892237	1897840	1927068	1932486
## 5	814686	832216	862276	918851	990473	1031241	1092223	1123586
## 6	1272295	1359494	1437573	1478469	1499946	1570269	1675350	1808303
## 7	817462	869037	905738	941553	977306	1057933	1148001	1196226
## 8	1290892	1338839	1393338	1462655	1559789	1695823	1863607	1987827
## 9	425463	426081	425141	427037	440011	466962	522475	463261
## 10	2329518	2562224	2782784	2981844	3187076	3462294	3677793	3703519
## 11	1671286	1767195	1849851	1794725	1729981	1710877	1800749	1919386
## 12	2789866	2974241	3098582	3217399	3414456	3439620	3540957	3383157
## 13	1275763	1415209	1512936	1560270	1650310	1769284	1861259	1896759
## 14	1999946	1892398	1744883	1697941	1807153	2012287	2178942	2286573
## 15	3485570	3418475	3337815	3368280	3608673	3959117	4392406	4852715
## 16	4694044	4742528	4841994	5072021	5406380	5698301	5906674	6040927
## 17	2180937	2363351	2612455	2840700	3010769	3172392	3319842	3341801
## 18	2149204	2143515	2143743	2235833	2489107	2972610	3528584	3822026
## 19	5001040	5036732	4978825	4829437	4758296	4627044	4495533	4321843
## 20	2003911	2065476	2119362	2219760	2290412	2339138	2410467	2449891
## 21	421593	445840	463780	474705	475365	466186	439092	418940
## 22	1281724	1328175	1350451	1316698	1288052	1253778	1254784	1255823
## 23	3438646	3554513	3615083	3632702	3595157	3470817	3614111	3679899
## 24	2136599	2112917	2055674	2014110	2032209	2054382	2049364	1969955
## 25	2073500	2038185	2000076	1941244	1901616	1905763	1930879	1921251
## 26	138992	143187	139920	135995	127882	120706	109148	92060
## 27	1643225	1802742	1941810	2037632	2118803	2139326	2192280	2212732
## 28	1384027	1410974	1420194	1422734	1444489	1377671	1344797	1306024
## 29	219463	214883	207231	219542	224954	226991	223162	214090
## 30	306755	298922	291970	283505	290375	280950	269277	268948
## 31	1003278	1027697	1064849	1115640	1170181	1153519	1158801	1133353
##	X2001	X2000	X1999					
## 1	525844	512351	473422					
## 2	450938	446430	428334					
## 3	4213686	4117058	3844420					
## 4	1740518	1660337	1590282					
## 5	1085032	1040857	995456					
## 6	1859905	1737624	1562444					
## 7	1201895	1168514	1094484					
## 8	2158178	2158672	2046920					
## 9	557948	562129	539780					
## 10	3246164	2934647	2666953					
## 11	2028231	1941184	1786620					
## 12	3083280	3041774	2944687					
## 13	1942569	1962632	1973559					

```

## 14 2246158 2206902 2146164
## 15 5780665 5699640 5264908
## 16 5886532 5629942 5077959
## 17 2997821 2810909 2587432
## 18 3503279 3291607 3028566
## 19 4054225 3881614 3797987
## 20 2450923 2487034 2360700
## 21 377803 375932 359666
## 22 1284945 1267003 1095743
## 23 3587236 3356703 2856294
## 24 1601444 1380966 1228515
## 25 1739587 1637575 1480201
## 26 56344 43121 34756
## 27 2016476 1877944 1693914
## 28 1183272 1085210 984753
## 29 188597 175269 160155
## 30 257889 250605 237254
## 31 1003155 934273 873343

NCEE_function <- function(JHS, reg) {
  intercept <- rep(NA, 31)
  slope <- rep(NA, 31)
  cor <- rep(NA, 31)
  predict_year <- c("2023", "2022", "2021", "2020", "2019")
  prd_df <- data.frame(predict_year)
  prd_low <- data.frame(predict_year)
  prd_high <- data.frame(predict_year)

  for (i in 1:31) {
    city_JHS <- as.numeric(JHS[i, 7:18])/30000 # 2012-2001
    city_reg <- as.numeric(reg[i, 1:12])/10000 # 2018-2007
    city_lm.i <- lm(city_reg ~ city_JHS)

    intercept[i] <- as.numeric(city_lm.i$coefficients[1])
    slope[i] <- as.numeric(city_lm.i$coefficients[2])
    cor[i] <- cor(city_JHS, city_reg)

    predict_city <- predict(city_lm.i, newdata = data.frame(city_JHS = as.numeric(NCEE_JHS[i,
      2:6])/30000))
    predict_city_low <- predict(city_lm.i, newdata = data.frame(city_JHS = as.numeric(NCEE_JHS[i,
      2:6])/30000), interval = "confidence")[, 2]
    predict_city_high <- predict(city_lm.i, newdata = data.frame(city_JHS = as.numeric(NCEE_JHS[i,
      2:6])/30000), interval = "confidence")[, 3]

    prd_df <- data.frame(prd_df, predict_city)
    prd_low <- data.frame(prd_low, predict_city_low)
    prd_high <- data.frame(prd_high, predict_city_high)
  }

  colnames(prd_df) <- c("year", province)
  colnames(prd_low) <- c("year", province)
  colnames(prd_high) <- c("year", province)

```

```

    summ <- data.frame(province, intercept, slope, cor)

    return(list(summ, prd_df, prd_low, prd_high))
}

# Sum up all province prediction as the national total
predict_national <- rep(NA, 5)
for (i in 1:5) {
  predict_national[i] <- sum(as.numeric(NCEE_function(NCEE_JHS, NCEE_reg)[[2]][i,
    ]))
} # 2023-2019

# =====Use National JHS as Predictors===== #

JHS_total <- NCEE_national$JHS[7:17]/30000
Reg_total <- NCEE_national$Reg[1:11]
new.data <- NCEE_national$JHS[2:6]/30000

test <- data.frame(JHS_total, Reg_total)

predict_national2 <- predict(lm(Reg_total ~ JHS_total, data = test), newdata = data.frame(JHS_total = new.data$JHS_total,
  interval = "confidence"))
predict_national

## [1] 838.4952 830.3795 831.1387 839.2643 843.9974
predict_national2

##          fit          lwr          upr
## 1 915.4487 847.6075 983.2899
## 2 912.1054 839.5374 984.6734
## 3 911.5534 838.1994 984.9074
## 4 910.7361 836.2158 985.2563
## 5 915.0885 846.7411 983.4360

# =====Plot prediction result===== #

national_reg <- c(rep(0, 5), NCEE_national$Reg[1:14])
national_JHS <- NCEE_national$JHS[2:20]/30000
year_born <- c(2005:1987)
predict_national_t <- c(rep(NA, 19), predict_national, national_reg[6:19])

predict_national_pf <- c(rep(NA, 19), prd_fit <- predict_national2[, 1], national_reg[6:19])
predict_national_pl <- c(rep(NA, 19), prd_fit <- predict_national2[, 2], rep(NA,
  14))
predict_national_ph <- c(rep(NA, 19), prd_fit <- predict_national2[, 3], rep(NA,
  14))

population <- c(national_JHS, national_reg)
label <- c(rep("JHS", 19), rep("NCEE Reg", 19))
year_birth <- rep(year_born, 2)

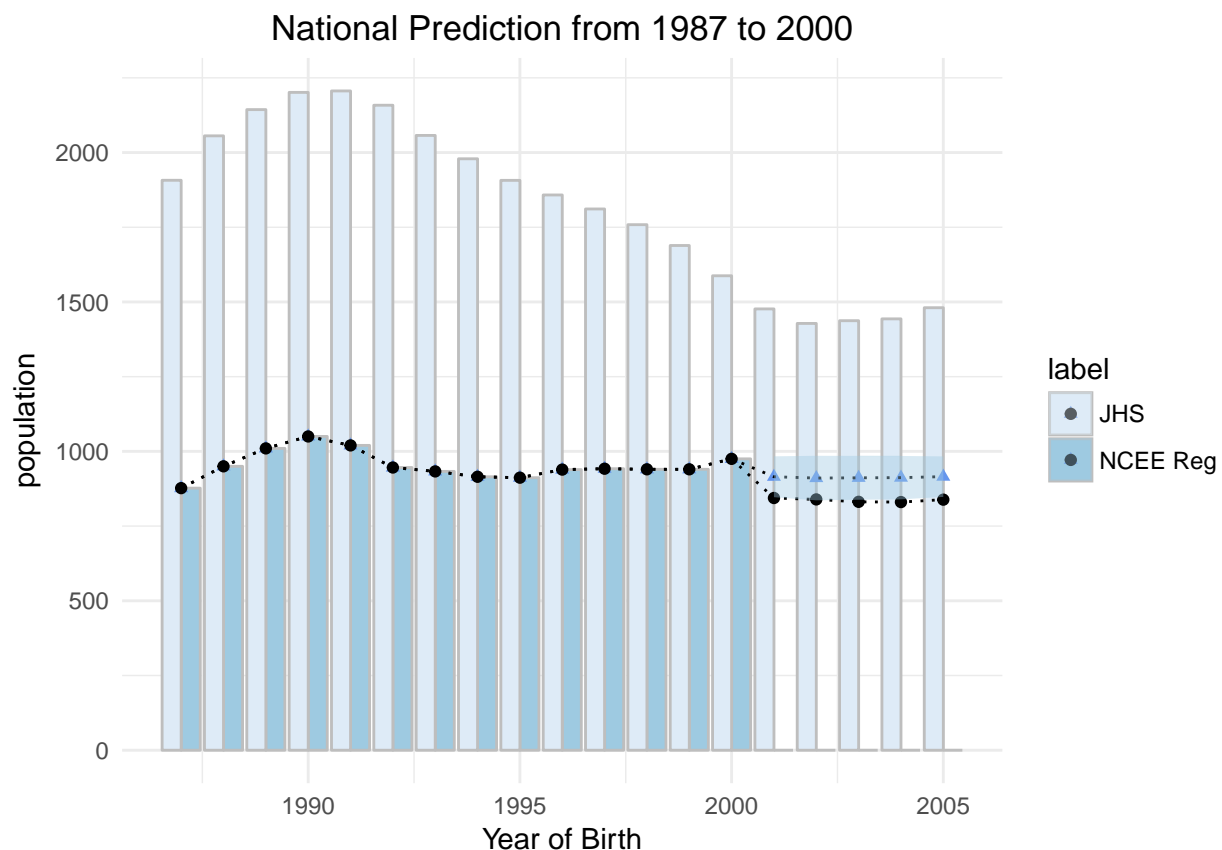
plot_df <- data.frame(population, label, year_birth, predict_national_t, predict_national_pf,
  predict_national_pl, predict_national_ph)

```

```
ggplot(data = plot_df, aes(x = year_birth, y = population, fill = label)) +
  geom_bar(stat = "identity", color = "grey", position = position_dodge()) +
  geom_point(aes(x = year_birth, y = predict_national_pf), color = "cornflowerblue",
    shape = 17) + geom_point(aes(x = year_birth, y = predict_national_t)) +
  geom_line(aes(x = year_birth, y = predict_national_pf), linetype = "dotted") +
  geom_line(aes(x = year_birth, y = predict_national_t), linetype = "dotted") +
  geom_ribbon(aes(ymin = predict_national_pl, ymax = predict_national_ph,
    x = year_birth), linetype = 2, alpha = 0.4) + theme_minimal() + scale_fill_brewer(palette = "Bl
  ggtitle("National Prediction from 1987 to 2000") + theme(plot.title = element_text(hjust = 0.5)) +
  xlab("Year of Birth")
```

```
## Warning: Removed 19 rows containing missing values (geom_point).
```

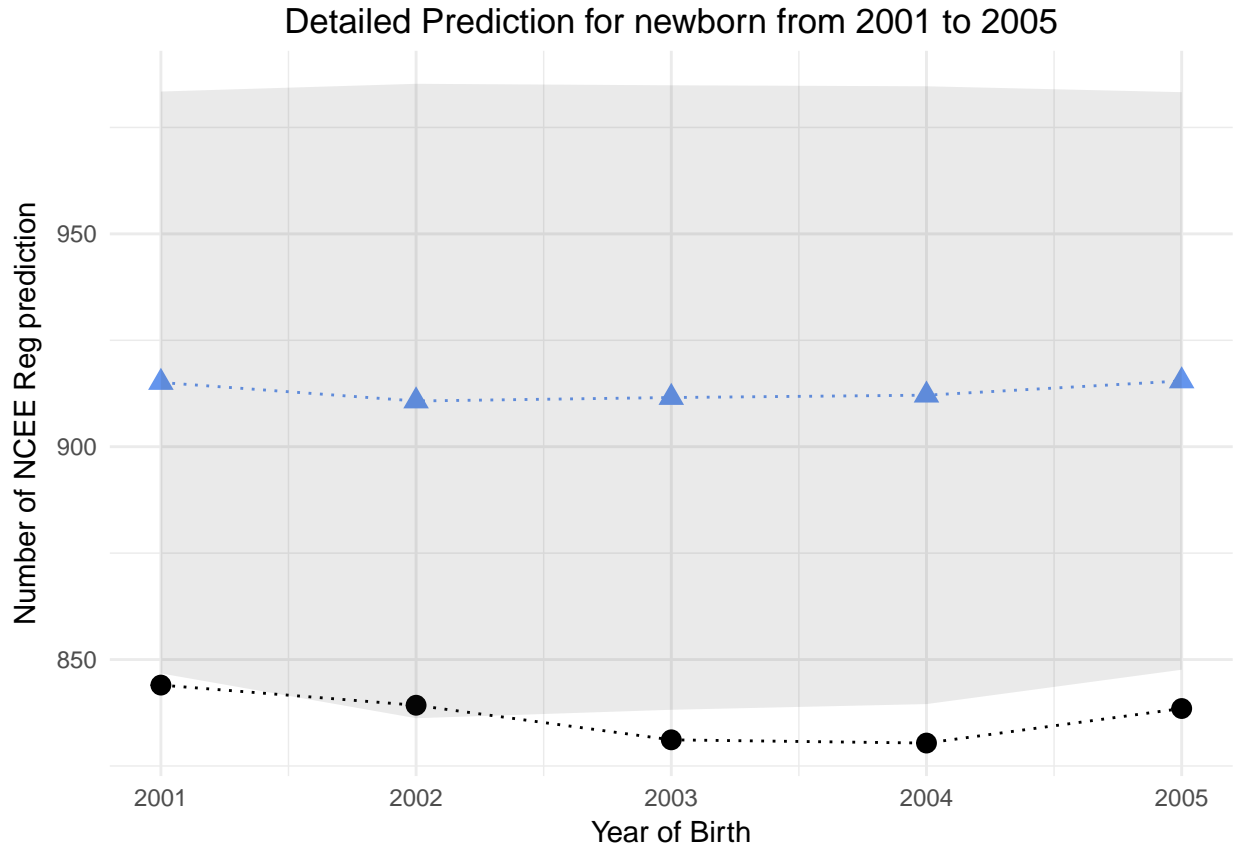
```
## Warning: Removed 19 rows containing missing values (geom_point).
```



```
new_born <- c(2005:2001)
new_plot <- data.frame(new_born, predict_national, predict_national2[, 1], predict_national2[,
  2], predict_national2[, 3])

ggplot(data = new_plot) + geom_line(aes(x = new_born, y = predict_national2[,
  1]), linetype = "dotted", color = "cornflowerblue") + geom_line(aes(x = new_born,
  y = predict_national), linetype = "dotted") + geom_point(aes(x = new_born,
  y = predict_national2[, 1]), color = "cornflowerblue", shape = 17, size = 3) +
  geom_point(aes(x = new_born, y = predict_national), size = 3) + geom_ribbon(aes(ymin = predict_nati
  2], ymax = predict_national2[, 3], x = new_born), linetype = 2, alpha = 0.1) +
  xlab("Year of Birth") + ylab("Number of NCEE Reg prediction") + ggtitle("Detailed Prediction for new")
```

```
theme_minimal() + theme(plot.title = element_text(hjust = 0.5))
```



```
city.low <- NCEE_function(NCEE_JHS, NCEE_reg)[[3]]
city.low.nu <- as.vector(t(city.low[, 2:32]))
city.fit <- NCEE_function(NCEE_JHS, NCEE_reg)[[2]]
city.fit.nu <- as.vector(t(city.fit[, 2:32]))
city.high <- NCEE_function(NCEE_JHS, NCEE_reg)[[4]]
city.high.nu <- as.vector(t(city.high[, 2:32]))

year_prd <- c(rep("23", 31), rep("22", 31), rep("21", 31), rep("20", 31), rep("19",
  31))

pro_bind <- data.frame(year_prd, province = rep(province, 5), city.low.nu, city.fit.nu,
  city.high.nu)

pl1 <- pro_bind %>% subset(province %in% c("Beijing", "Tianjin", "Shanghai",
  "Hainan")) %>% ggplot(aes(year_prd, city.fit.nu, color = province)) + geom_pointrange(aes(ymin = ci
  ymax = city.high.nu)) + scale_color_manual(values = wes_palette("Moonrise2",
  4)) + geom_line(aes(group = province)) + theme_minimal() + xlab("Year") +
  ylab("Number of Reg Prediction") + ggtitle("3 Metropolis & Hainan Island") +
  theme(plot.title = element_text(hjust = 0.5))

pl2 <- pro_bind %>% subset(province %in% c("Qinghai", "Ningxia", "Xizang")) %>%
  ggplot(aes(year_prd, city.fit.nu, color = province)) + geom_pointrange(aes(ymin = city.low.nu,
  ymax = city.high.nu)) + scale_color_manual(values = wes_palette("Rushmore1",
  3)) + geom_line(aes(group = province)) + theme_minimal() + xlab("Year") +
```



```

    ylab("Number of Reg Prediction") + ggtitle("West 4 provinces") + theme(plot.title = element_text(hj
p13 <- pro_bind %>% subset(province %in% c("Heilongjiang", "Jilin", "Liaoning",
    "Neimenggu")) %>% ggplot(aes(year_prd, city.fit.nu, color = province)) +
    geom_pointrange(aes(ymin = city.low.nu, ymax = city.high.nu)) + scale_color_manual(values = wes_pal
4)) + geom_line(aes(group = province)) + theme_minimal() + xlab("Year") +
    ylab("Number of Reg Prediction") + ggtitle("North-East 4 provinces") + theme(plot.title = element_t

p14 <- pro_bind %>% subset(province %in% c("Sichuan", "Guizhou", "Yunnan", "Guangxi",
    "Chongqing")) %>% ggplot(aes(year_prd, city.fit.nu, color = province)) +
    geom_pointrange(aes(ymin = city.low.nu, ymax = city.high.nu)) + scale_color_manual(values = wes_pal
5)) + geom_line(aes(group = province)) + theme_minimal() + xlab("Year") +
    ylab("Number of Reg Prediction") + ggtitle("South-West 5 provinces") + theme(plot.title = element_t

p15 <- pro_bind %>% subset(province %in% c("Zhejiang", "Jiangsu", "Anhui")) %>%
    ggplot(aes(year_prd, city.fit.nu, color = province)) + geom_pointrange(aes(ymin = city.low.nu,
    ymax = city.high.nu)) + scale_color_manual(values = wes_palette("FantasticFox1",
3)) + geom_line(aes(group = province)) + theme_minimal() + xlab("Year") +
    ylab("Number of Reg Prediction") + ggtitle("South-East cost 3 provinces") +
    theme(plot.title = element_text(hjust = 0.5))

p16 <- pro_bind %>% subset(province %in% c("Guangdong", "Fujian", "Jiangxi")) %>%
    ggplot(aes(year_prd, city.fit.nu, color = province)) + geom_pointrange(aes(ymin = city.low.nu,
    ymax = city.high.nu)) + scale_color_manual(values = wes_palette("GrandBudapest1",
3)) + geom_line(aes(group = province)) + theme_minimal() + xlab("Year") +
    ylab("Number of Reg Prediction") + ggtitle("South-Coast 3 provinces") +
    theme(plot.title = element_text(hjust = 0.5))

p17 <- pro_bind %>% subset(province %in% c("Hunan", "Hubei", "Anhui", "Henan",
    "Shandong")) %>% ggplot(aes(year_prd, city.fit.nu, color = province)) +
    geom_pointrange(aes(ymin = city.low.nu, ymax = city.high.nu)) + scale_color_manual(values = wes_pal
5)) + geom_line(aes(group = province)) + theme_minimal() + xlab("Year") +
    ylab("Number of Reg Prediction") + ggtitle("North-Central 5 provinces") +
    theme(plot.title = element_text(hjust = 0.5))

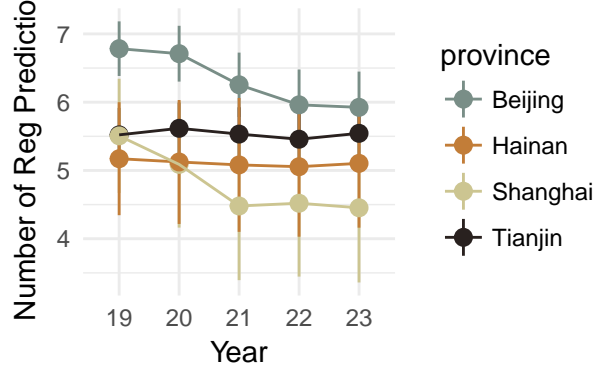
p18 <- pro_bind %>% subset(province %in% c("Shanxi", "Shaanxi", "Hebei", "Gansu")) %>%
    ggplot(aes(year_prd, city.fit.nu, color = province)) + geom_pointrange(aes(ymin = city.low.nu,
    ymax = city.high.nu)) + scale_color_manual(values = wes_palette("Rushmore",
4)) + geom_line(aes(group = province)) + theme_minimal() + xlab("Year") +
    ylab("Number of Reg Prediction") + ggtitle("West-Central 4 provinces") +
    theme(plot.title = element_text(hjust = 0.5))

# =====Display===== #

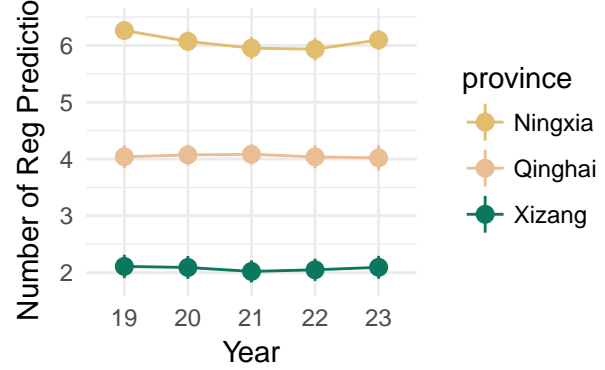
grid.arrange(pl1, pl2, pl3, pl4, nrow = 2)

```

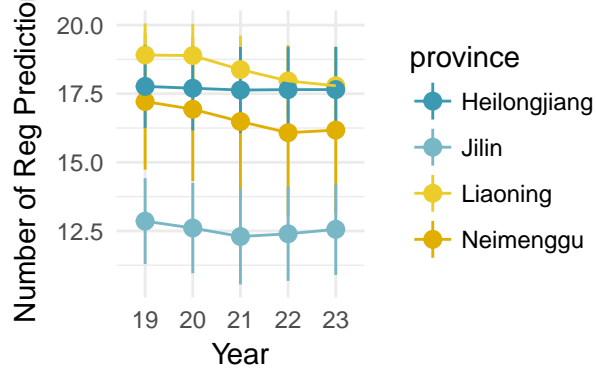
Metropolis & Hainan Island



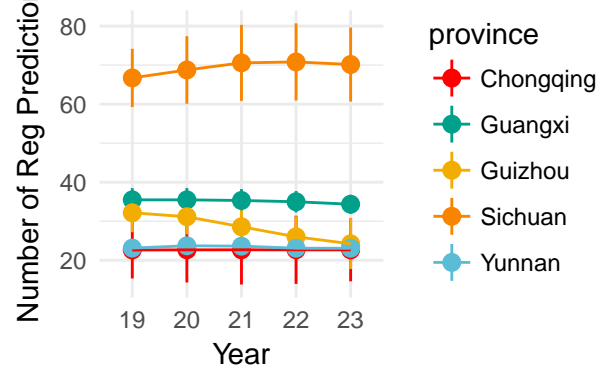
West 4 provinces



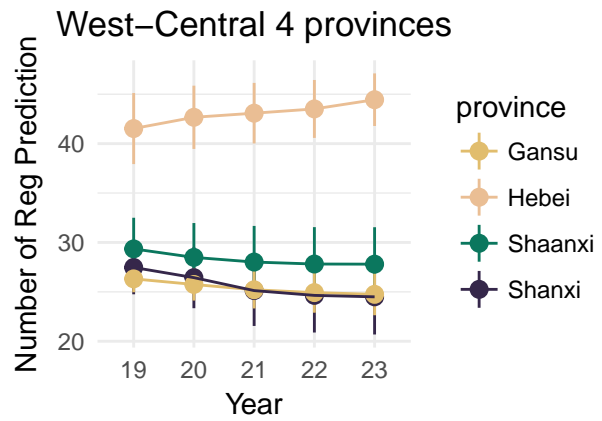
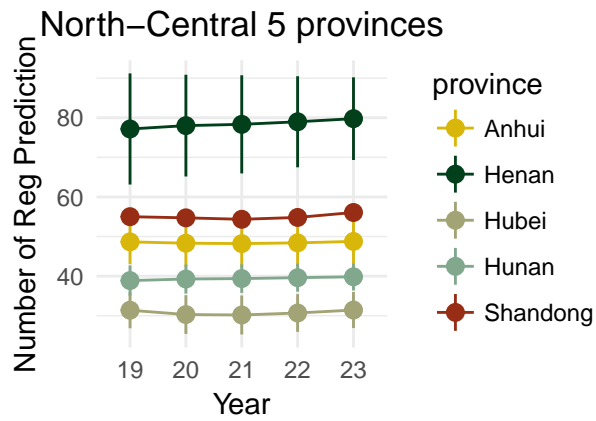
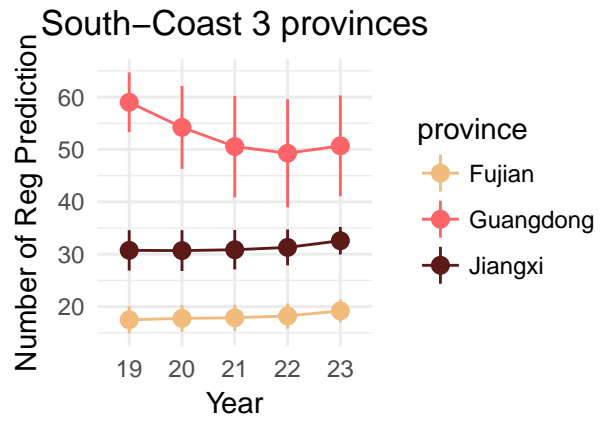
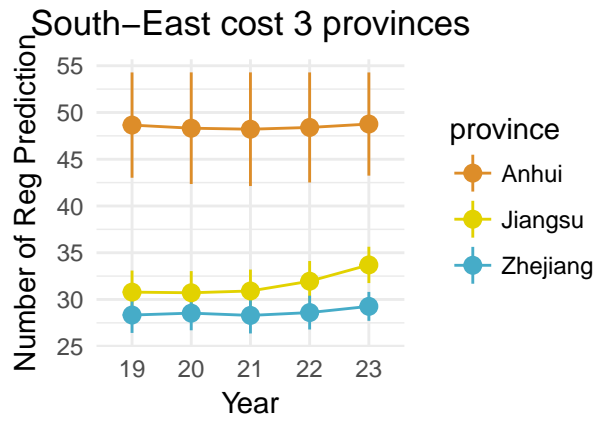
North-East 4 provinces



South-West 5 provinces



```
grid.arrange(p15, p16, p17, p18, nrow = 2)
```



```
# ggplot(map('china', plot=F), aes(long, lat, group=group, fill=region)) +
# geom_path(show.legend = F) + ggtitle('Map of China') + geom_polygon()

# chinamap <- readRDS('gadm36_CHN_0_sp.rds')

# ggplot(chinamap, aes(long, lat, group=group)) + geom_polygon(fill='white',
# # colour='gray') + ggtitle('Map of China')
```