

# Study Visas

Analyse student visa dataset provided by Immigration New Zealand.

```
study.df <- read.csv('S7 - First-time students (data).csv', sep=',', header=T)

# fix up datatypes
#fy.date <- function(dt) {
#  lapply(as.numeric(strsplit(dt, '/')[[1]]))
#}

# FY.XXX is of the form LastYear/CurrentYear, e.g. 2014/15. We'll use the first part: 2014
#study.df$FY.Decided <- sapply(as.character(study.df$FY.Decided), fy.date)
#study.df$FY.Resident.Decision <- sapply(as.character(study.df$FY.Resident.Decision), fy.date)
#study.df$FY.Work.Decision <- sapply(as.character(study.df$FY.Work.Decision), fy.date)

study.df$Month.Decision <- as.Date(paste(study.df$Month.Decision, "-01", sep=""), "%Y-%m-%d")

str(study.df)

## 'data.frame': 80003 obs. of 19 variables:
## $ Number.of.Clients : int 3 2 1 1 1 1 1 7 1 ...
## $ FY.Decided : Factor w/ 6 levels "2010/11","2011/12",...: 1 1 1 1 1 1 1 1 1 ...
## $ Month.Decision : Date, format: "2010-07-01" "2010-07-01" ...
## $ Decision.Location : Factor w/ 2 levels "Offshore","Onshore": 1 1 1 1 1 1 1 1 1 ...
## $ Application.Criteria: Factor w/ 17 levels "Dependant of a Diplomat",...: 2 2 2 2 2 2 2 2 2 ...
## $ Decision.Type : Factor w/ 1 level "Approved": 1 1 1 1 1 1 1 1 1 ...
## $ Student.Type : Factor w/ 4 levels "Dependant","Fee Paying",...: 1 1 1 1 1 1 1 1 1 ...
## $ Agent.Type : Factor w/ 6 levels "Adviser (Exempt)",...: 2 6 6 6 6 6 6 6 6 ...
## $ Nationality : Factor w/ 184 levels "Afghanistan",...: 130 23 29 29 59 100 112 127 130 130 ...
## $ Nationality.Top.10 : Factor w/ 11 levels "Brazil","China",...: 8 1 7 7 4 7 7 7 8 8 ...
## $ Institution.Type : Factor w/ 14 levels "(not recorded)",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ Location : Factor w/ 162 levels "", "(Christchurch, Tauranga, Queenstown)",...: 102 103 ...
## $ Arrival.Recorded. : Factor w/ 2 levels "Arrival recorded",...: 1 1 1 1 1 2 1 1 1 ...
## $ Resident.Approved. : Factor w/ 2 levels "No resident application approved",...: 2 1 1 2 2 1 2 1 2 ...
## $ FY.Resident.Decision: Factor w/ 17 levels ".", "1997/98",...: 13 1 1 13 12 1 14 1 12 13 ...
## $ Resident.Criteria : Factor w/ 57 levels "(not recorded)",...: 52 1 1 52 52 1 52 1 52 52 ...
## $ Work.Approved. : Factor w/ 2 levels "No work application approved",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ FY.Work.Decision : Factor w/ 20 levels ".", "1997/98",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ Work.Criteria : Factor w/ 90 levels "(not recorded)",...: 1 1 1 1 1 1 1 1 1 1 ...

kable(head(study.df), digits=2, caption='First-time students')
```

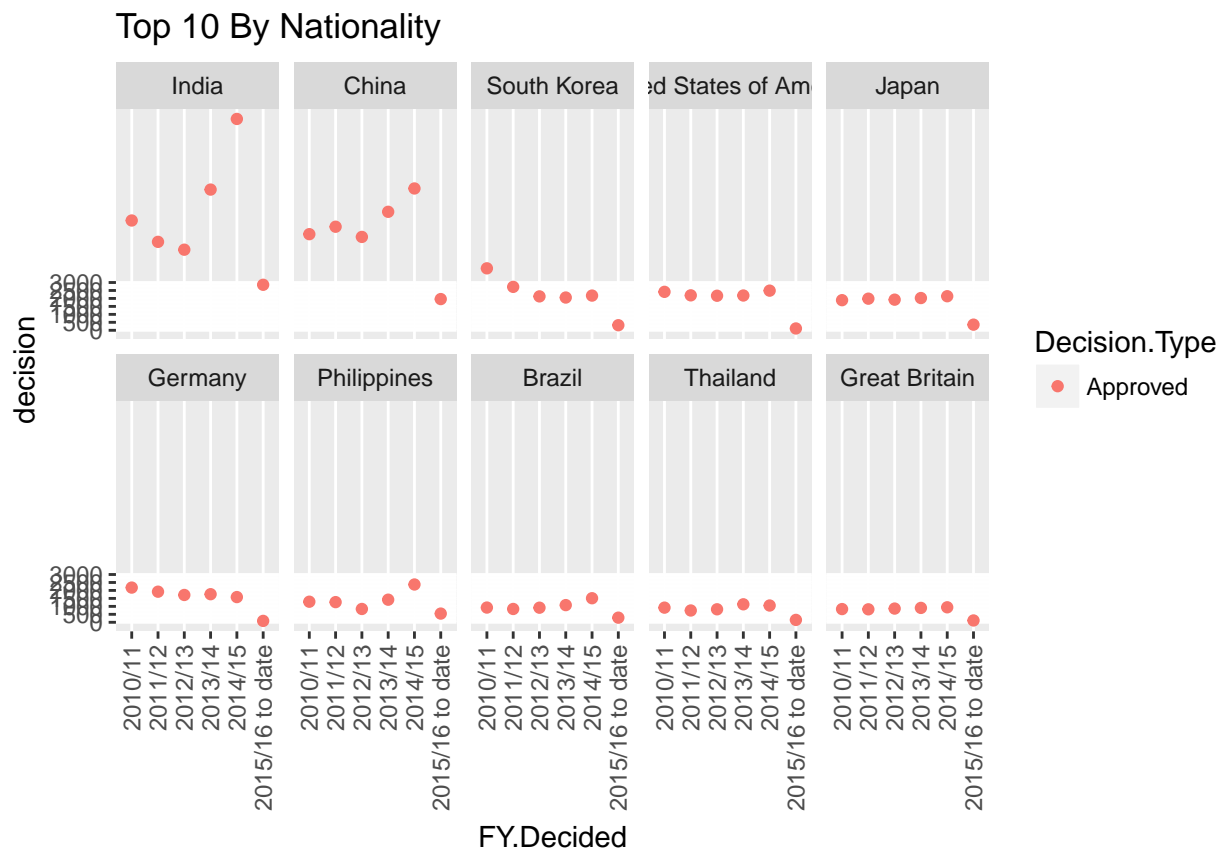
Number.of.Clients	FY.Decided	Month.Decision	Decision.Location	Application.Criteria	Decision.Type
3	2010/11	2010-07-01	Offshore	Dependant of a Worker - SMC	Approved
2	2010/11	2010-07-01	Offshore	Dependant of a Worker - SMC	Approved
1	2010/11	2010-07-01	Offshore	Dependant of a Worker - SMC	Approved
1	2010/11	2010-07-01	Offshore	Dependant of a Worker - SMC	Approved
1	2010/11	2010-07-01	Offshore	Dependant of a Worker - SMC	Approved
1	2010/11	2010-07-01	Offshore	Dependant of a Worker - SMC	Approved

## Number of First Students Over Time

```
by_nationality <- study.df %>%
  group_by(Nationality.Top.10, FY.Decided, Decision.Type) %>%
  summarise(decision=sum(Number.of.Clients)) %>%
  filter(Nationality.Top.10 != 'Other nationalities') %>%
  data.frame

# reorder factor by top country
nationality_order <- by_nationality %>% group_by(Nationality.Top.10) %>% summarise(decision=sum(decision))
by_nationality$Nationality.Top.10 <- factor(by_nationality$Nationality.Top.10, levels=nationality_order)

#plot it
ggplot(by_nationality, aes(x=FY.Decided, y=decision, colour=Decision.Type)) + geom_point() +
  facet_wrap(~Nationality.Top.10, ncol=5) +
  scale_y_continuous(minor_breaks=seq(0, 3000, 50), breaks=seq(0, 3000, 500)) +
  ggtitle("Top 10 By Nationality") +
  theme(axis.text.x = element_text(angle=90, hjust=1, vjust=0.5))
```



## Market Share

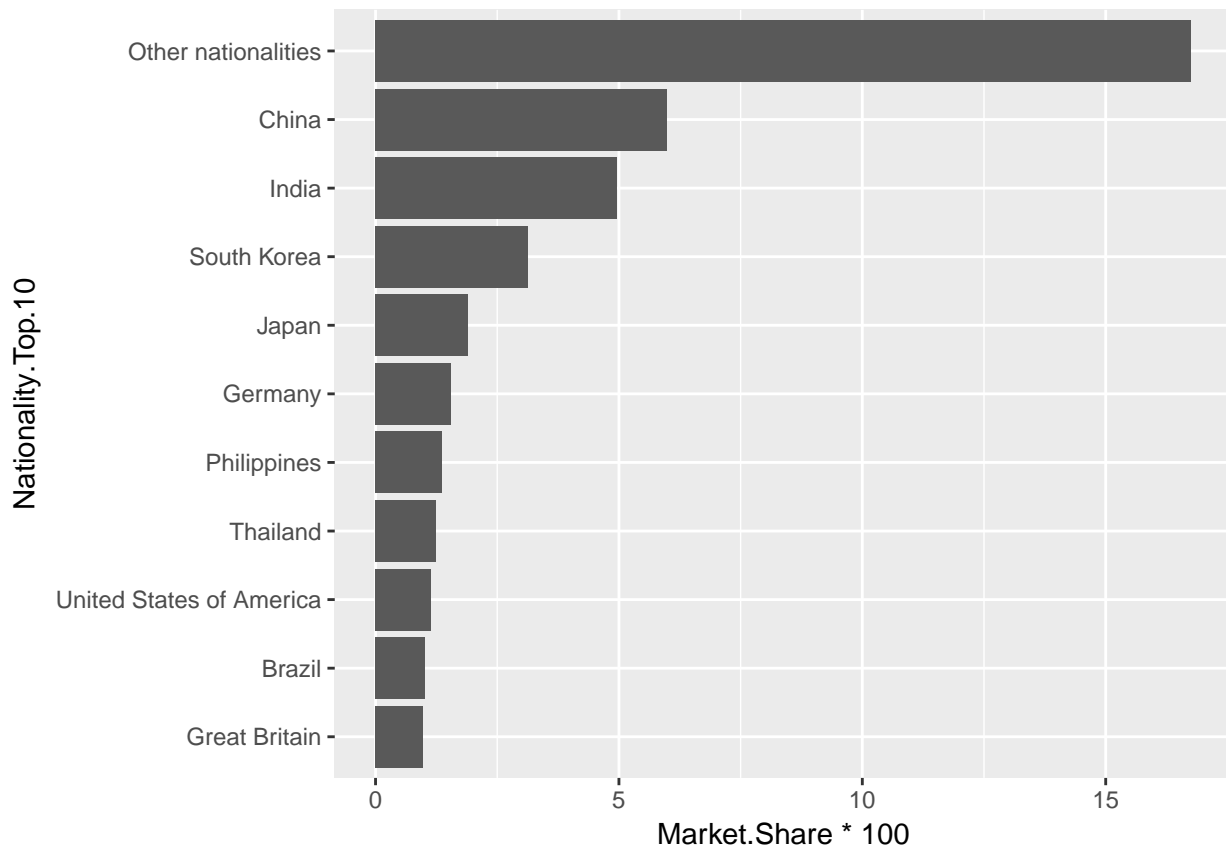
```
total <- sum(study.df$Number.of.Clients)
by_nationality <- study.df %>%
  group_by(Nationality.Top.10) %>%
  summarise(decision=n()) %>%
```

```

mutate(Market.Share=decision/total) %>%
data.frame
# reorder factor by top country
by_nationality$Nationality.Top.10 <- factor(by_nationality$Nationality.Top.10, levels=arrange(by_nationality$Market.Share, desc(Market.Share)))

# plot it
ggplot(by_nationality, aes(x=Nationality.Top.10, y=Market.Share * 100)) + geom_bar(stat='identity') +
  scale_y_continuous(breaks=seq(0, 100, 5)) + coord_flip()

```

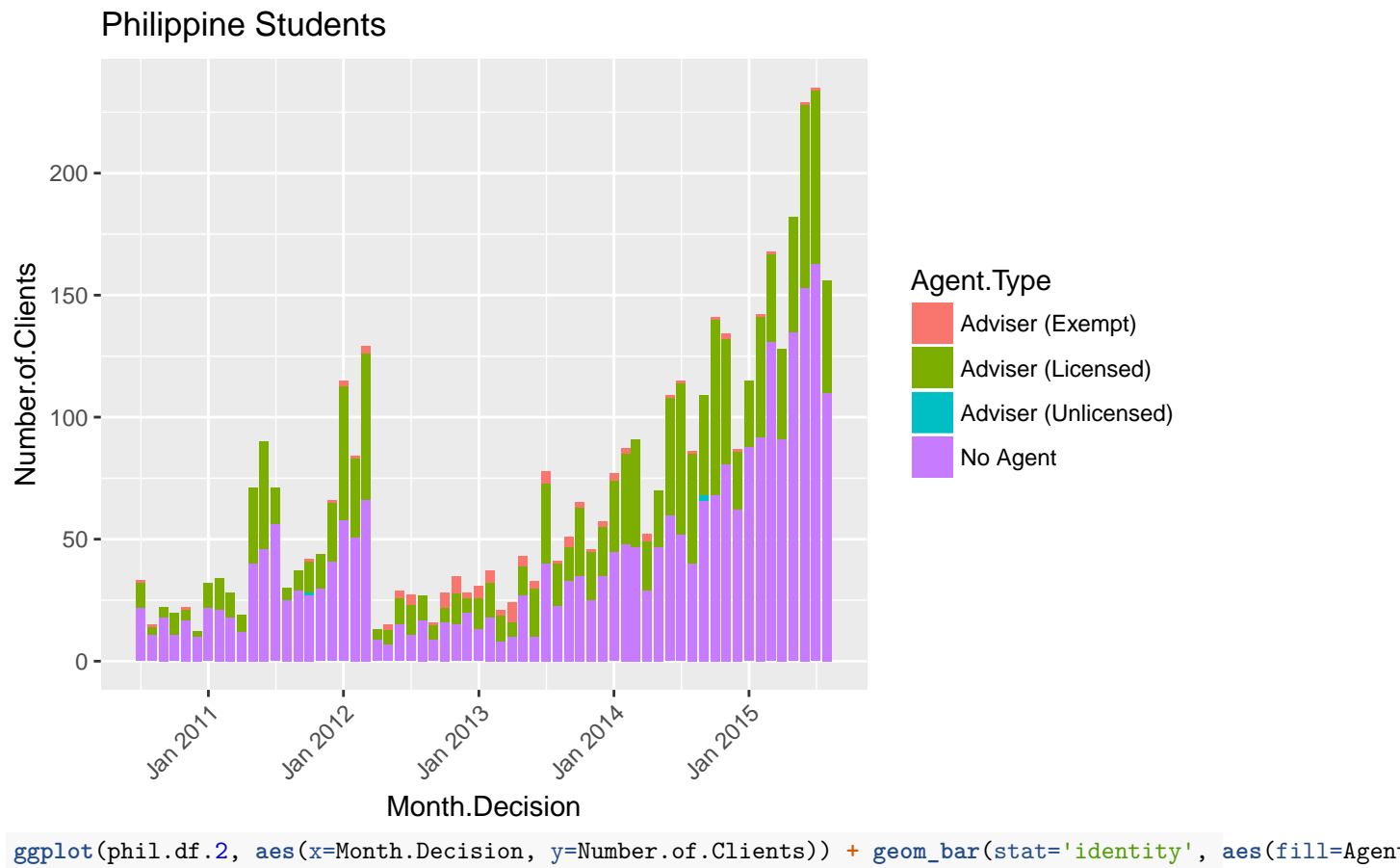


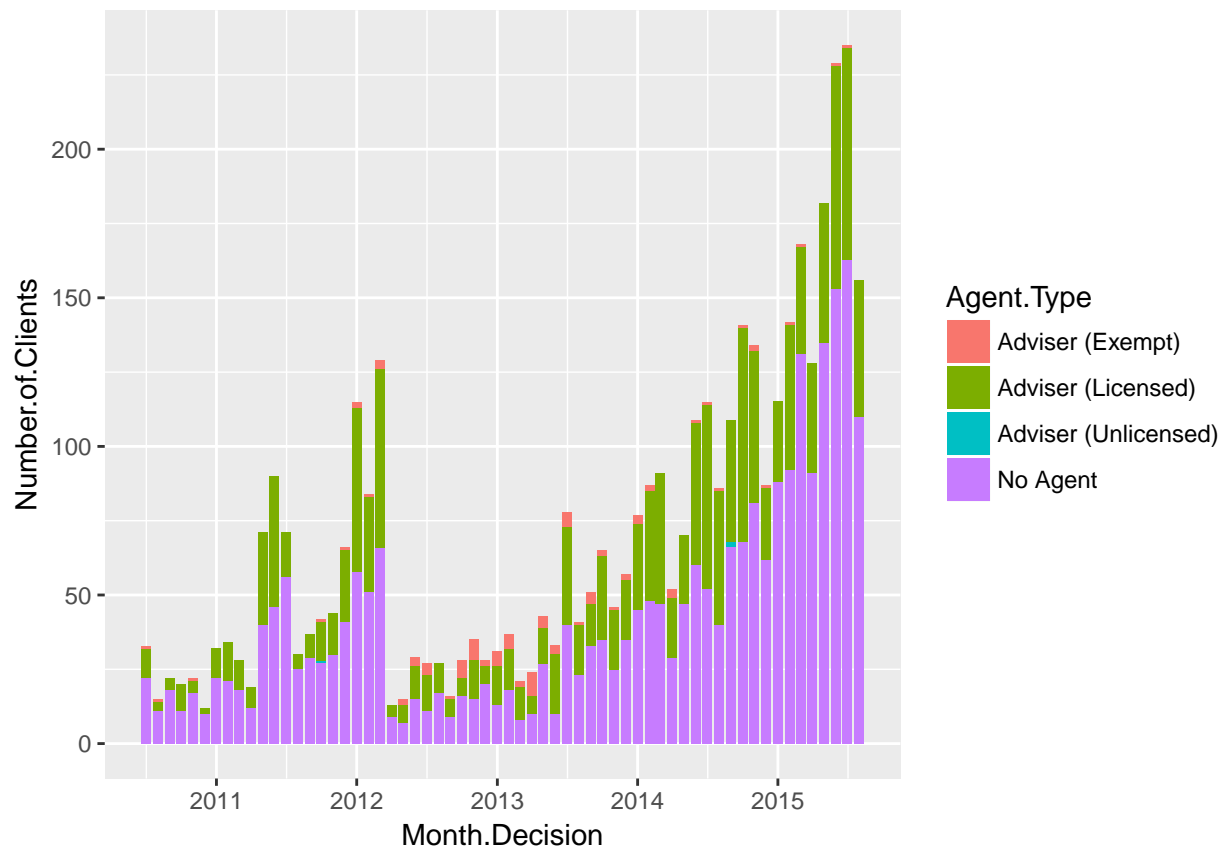
## Philippine Market

```

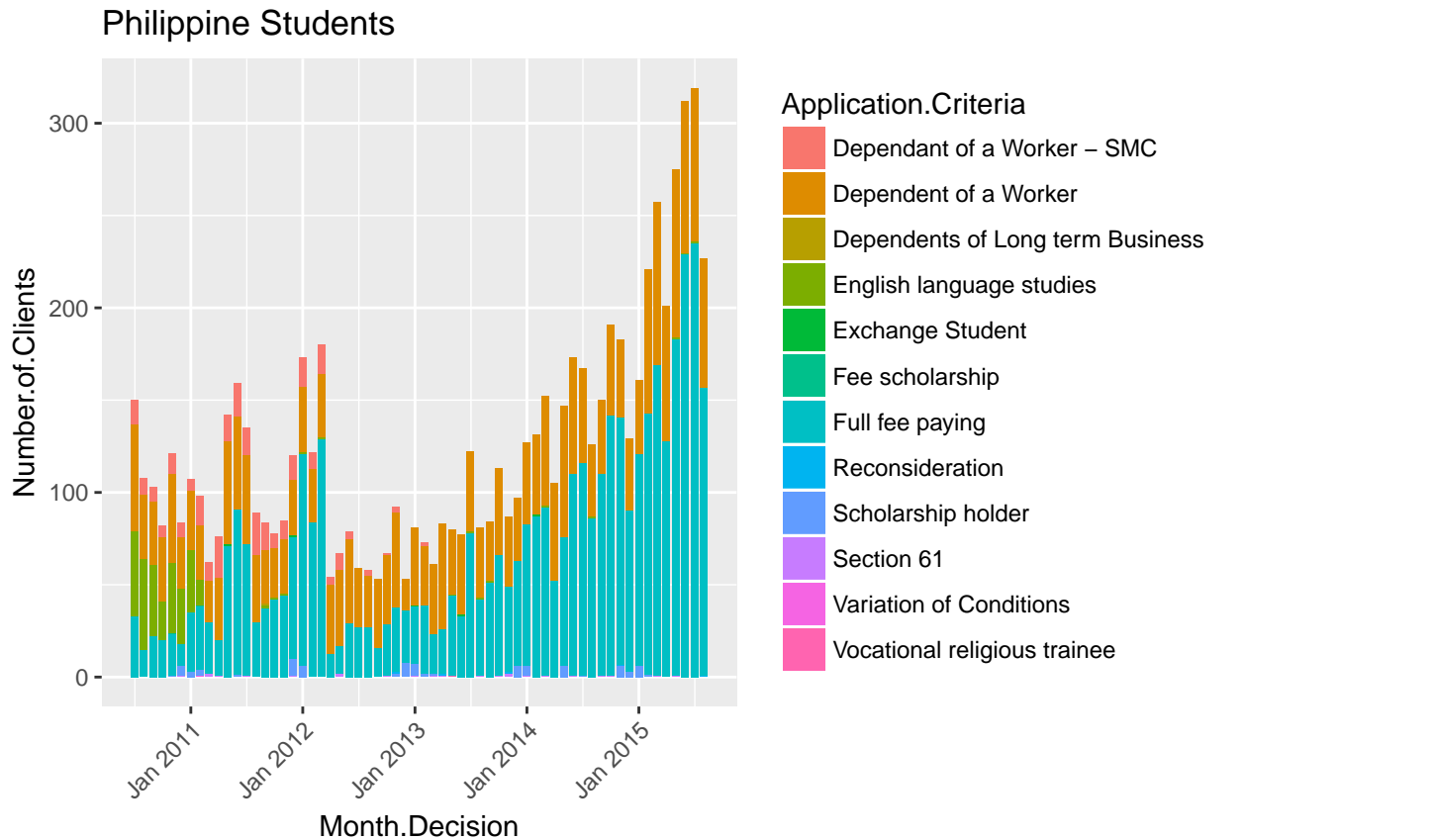
phil.df <- study.df %>% filter(Nationality == 'Philippines', Application.Criteria == 'Full fee paying')
phil.df.2 <- phil.df %>% group_by(Month.Decision, Agent.Type) %>%
  summarise(Number.of.Clients=sum(Number.of.Clients))
ggplot(phil.df.2, aes(x=Month.Decision, y=Number.of.Clients)) + geom_bar(stat='identity', aes(fill=Agent.Type)) +
  scale_x_date(labels=date_format("%b %Y")) + #, breaks="3 months", minor_breaks="1 month") +
  theme(axis.text.x=element_text(angle=45, hjust=1)) + ggtitle("Philippine Students")

```





```
phil.df <- study.df %>% filter(Nationality == 'Philippines')
phil.df.2 <- phil.df %>% group_by(Month.Decision, Application.Criteria) %>%
  summarise(Number.of.Clients=sum(Number.of.Clients))
ggplot(phil.df.2, aes(x=Month.Decision, y=Number.of.Clients)) +
  geom_bar(stat='identity', aes(fill=Application.Criteria)) +
  scale_x_date(labels=date_format("%b %Y")) + #, breaks="3 months", minor_breaks="1 month") +
  theme(axis.text.x=element_text(angle=45, hjust=1)) + ggtitle("Philippine Students")
```



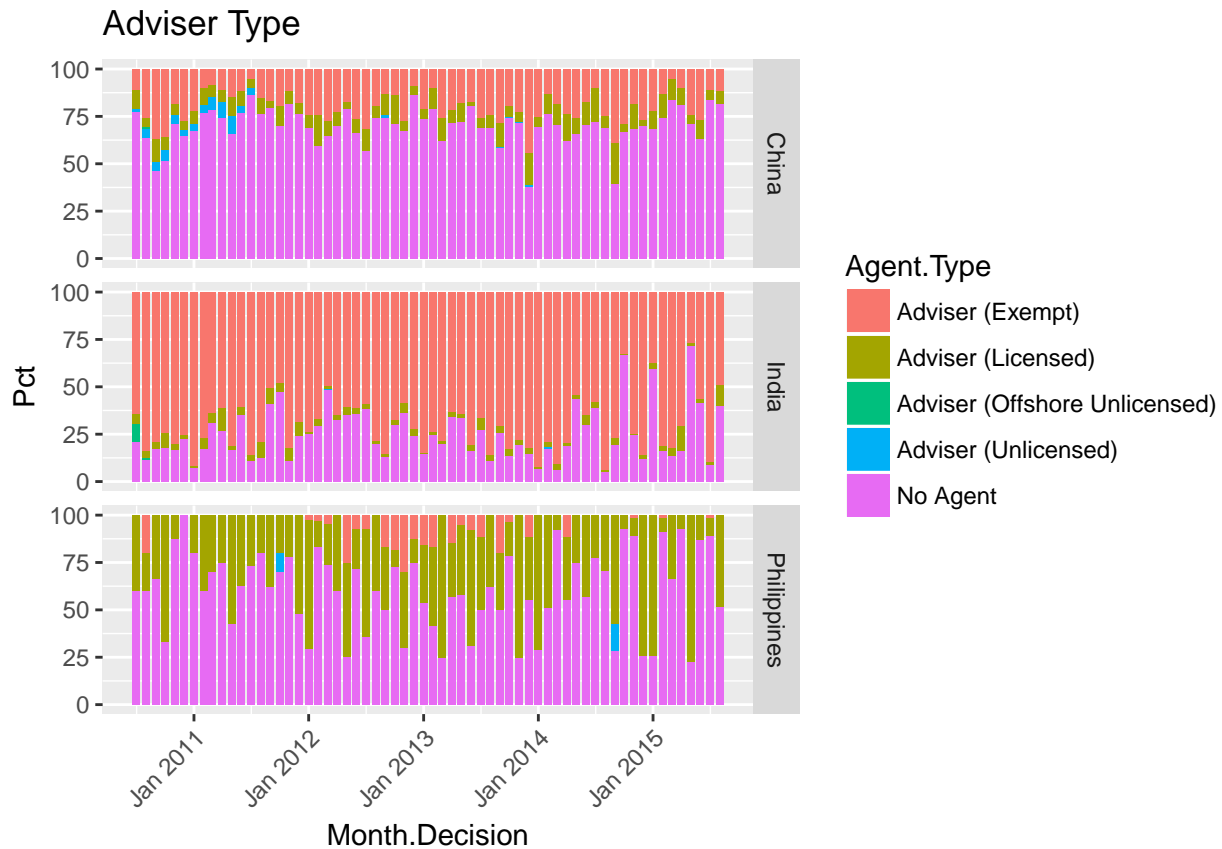
```
advisers.df <- study.df %>% filter(Nationality == c('Philippines', 'China', 'India'), Application.Crite

## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length

## Warning in `==.default`(Nationality, c("Philippines", "China", "India")):
## longer object length is not a multiple of shorter object length

country.totals <- group_by(advisers.df, Month.Decision, Nationality) %>% summarise(total=sum(Number.of.C
advisers.df <- left_join(advisers.df, country.totals, by=c("Nationality"="Nationality", "Month.Decision"
advisers.df.2 <- advisers.df %>% group_by(Month.Decision, Agent.Type, Nationality, total) %>%
  summarise(Number.of.Clients=sum(Number.of.Clients))

advisers.df.2 <- mutate(advisers.df.2, Pct=Number.of.Clients/total * 100)
ggplot(advisers.df.2, aes(x=Month.Decision, y=Pct)) + geom_bar(stat='identity', aes(fill=Agent.Type)) +
  scale_x_date(labels=date_format("%b %Y")) + #, breaks="3 months", minor_breaks="1 month") +
  scale_y_continuous() +
  facet_grid(Nationality~.) +
  theme(axis.text.x=element_text(angle=45, hjust=1)) + ggtitle("Adviser Type")
```

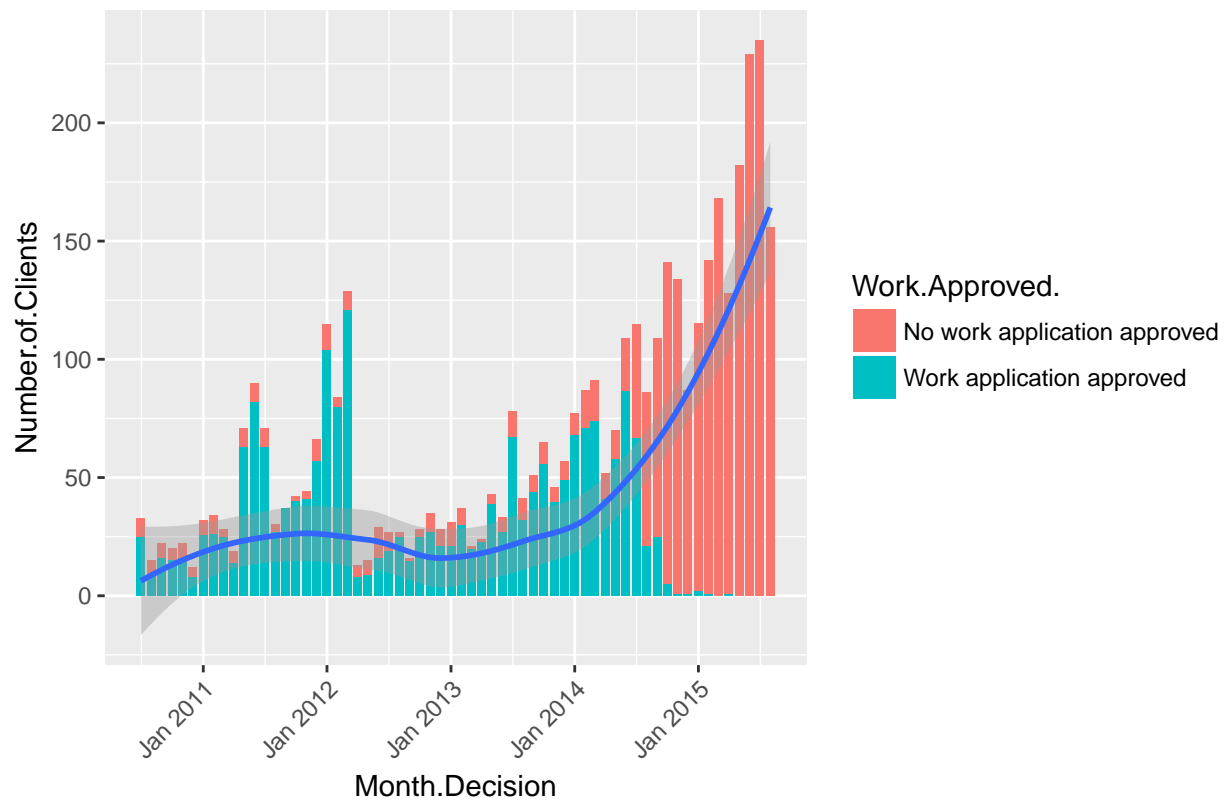


### By Agent Type - Work Visa Approved

```
phil.df <- study.df %>% filter(Nationality == 'Philippines', Application.Criteria == 'Full fee paying')
work.approved.df <- phil.df %>% group_by(Month.Decision, Work.Approved.) %>%
  summarise(Number.of.Clients=sum(Number.of.Clients))
ggplot(work.approved.df, aes(x=Month.Decision, y=Number.of.Clients)) +
  geom_bar(stat='identity', aes(fill=Work.Approved.)) +
  scale_x_date(labels=date_format("%b %Y")) + #, breaks="3 months", minor_breaks="1 month") +
  geom_smooth() +
  theme(axis.text.x=element_text(angle=45, hjust=1)) +
  ggtitle("Philippine Students - Work Visa Approval")

## `geom_smooth()` using method = 'loess'
```

## Philippine Students – Work Visa Approval



## By Agent Type - Residency Approved

```
residency.approved.df <- phil.df %>% group_by(Month.Decision, Resident.Approved.) %>%
  summarise(Number.of.Clients=sum(Number.of.Clients))
ggplot(residency.approved.df, aes(x=Month.Decision, y=Number.of.Clients)) +
  geom_bar(stat='identity', aes(fill=Resident.Approved.)) +
  scale_x_date(labels=date_format("%b %Y")) + #, breaks="3 months", minor_breaks="1 month") +
  theme(axis.text.x=element_text(angle=45, hjust=1)) +
  ggtitle("Philippine Students - Residence Visa Approval")
```



## Philippine Students – Residence Visa Approval



```
table(study.df$Decision.Type)
```

```
##
## Approved
##      80003
```

```
ggplot(study.df, aes(x=Application.Criteria)) + geom_bar() + coord_flip()
```

Application.Criteria

