Final Work

Contents

1. Part I : Short and Simple

1.1. AI Hype

- AI Hype comes with many advantages. In several years AI will dominate human life by forecasting best economic actions, proposing next best-action (in almost every life-action), early-diagnosing many ilnesses and optimizing production chains. In my opinion the most and the best outcome of AI is minimizing the human error. Especially in health sector automated and intelligent robots may practise surgeries that is currently imposible. This can really extend human-life. But there is a huge risk about massive unemployment.
- The human source is the key to develop AI. Well-educated engineers with high mental-capacity are required. In turkey however there is really huge capacity, current education system can not enchance the students with brand-new technologies. As a person that his brother working in America for 15 years, our country does not offer good opportunities. If we consider a sufficient level 100, turkey's level can only be 20. According to WIPO's official figure, AI patenting numbers there is no Turkish Company or Turkish University in the list. This is a huge and frustrating evidence.

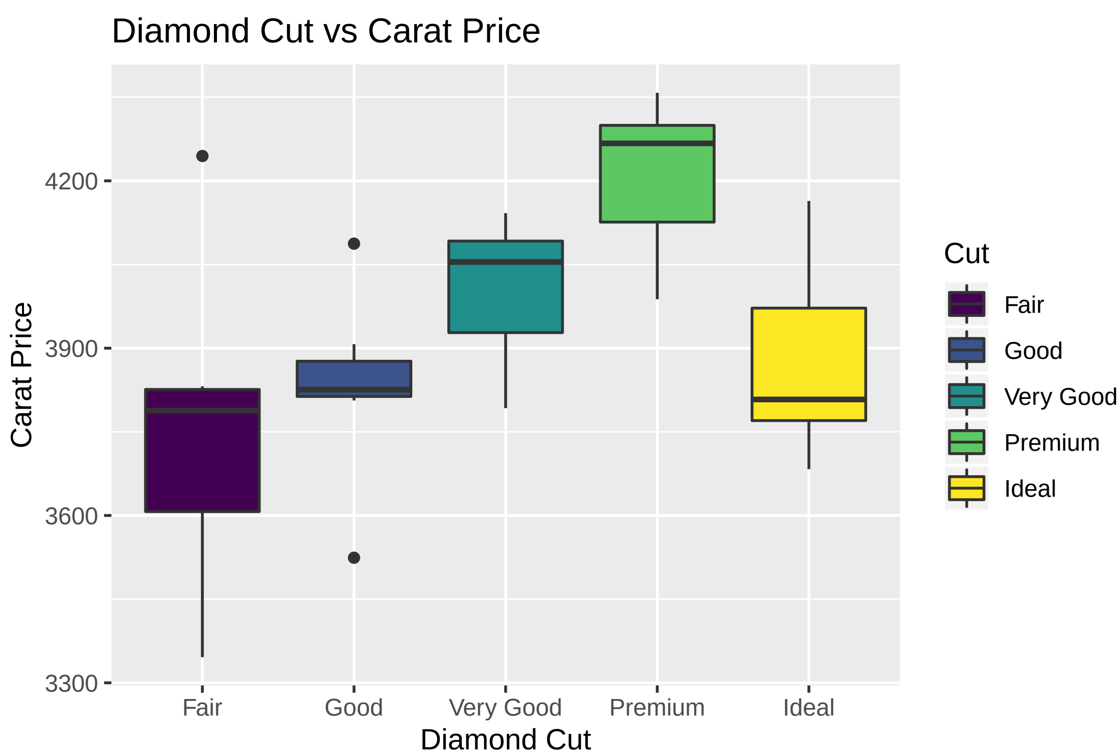
1.2. Exploratory Data Analysis Workflow

- My very first step to EDA is analyzing statistics of data (like mean, median, std) and detect outliers. By these statistics we can find data anomalies to fix. To detect ourliers box plotting is a very good tool. Then we can analyze variables. Categoric (ordinal/nominal) variables' frequencies and distribution, numerical (discrete/continous) variables' tendency may give an idea. Histograms and boxplotting is very common to analyze distributions. ggplot and plotly libraries area my favorites.
- In donations sample I assume that we have current category (like education level) levels and people distribution according to levels. Maximum number of human access could be my main goal. I think donations should be distributed based on this. Also lowest levelled categories can be prioritiezed.
- If I was more inclined for a policy, I would try to use data to support my thesis. I think we could find some guiding data for this perspective. Honesty would not be my priority. But if there is evidence that refuses my thesis, I am going to change it honestly(!)

1.3. Diamonds Analysis

- We see that cut property is very important for carat price.

```
my_diamond <- diamonds %>% mutate(carat_price=price/carat) %>% group_by(color, cut) %>%
  summarise(mean_carat_price =mean(carat_price)) %>% arrange(desc(mean_carat_price))
ggplot(my_diamond, aes(x=cut, y=mean_carat_price, fill=cut)) + geom_boxplot() +
  labs(x="Diamond Cut", y="Carat Price",title="Diamond Cut vs Carat Price", fill = "Cut")
```



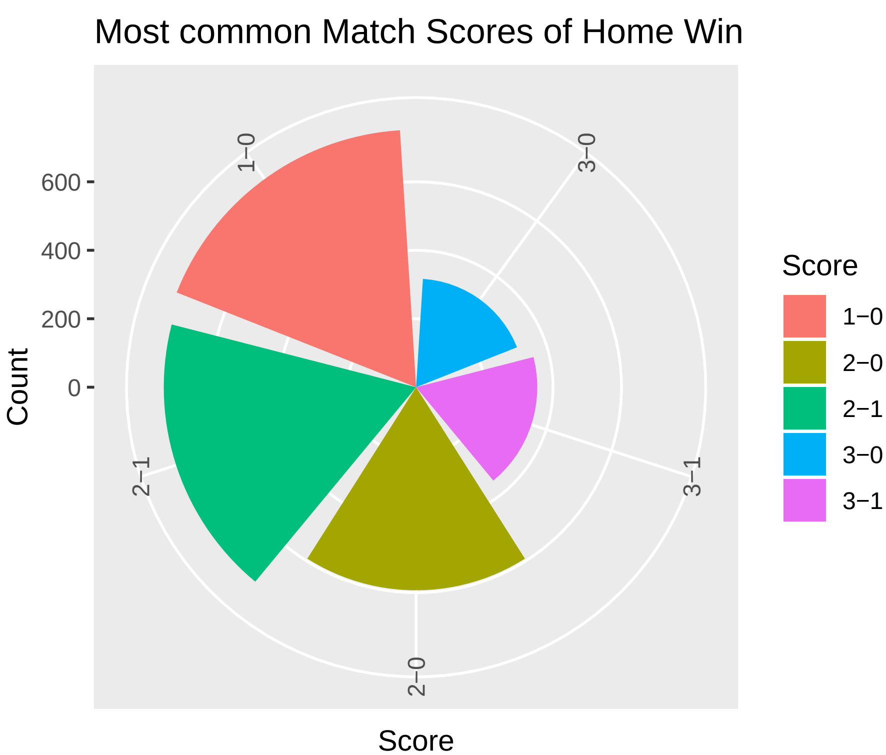
2. Part II: Extending Our Group Project

```
most_common_match_results <- raw_data %>% unite(match_score,c(FTHG, FTAG), sep="-") %>% select(season,
most_common_match_results <- most_common_match_results %>% group_by(match_result, match_score) %>%
  summarise(count=n()) %>% top_n(5, wt=count)

common_home_win <-most_common_match_results %>% filter(match_result == "H")
common_away_win <-most_common_match_results %>% filter(match_result == "A")
common_draw <- most_common_match_results %>% filter(match_result == "D")
```

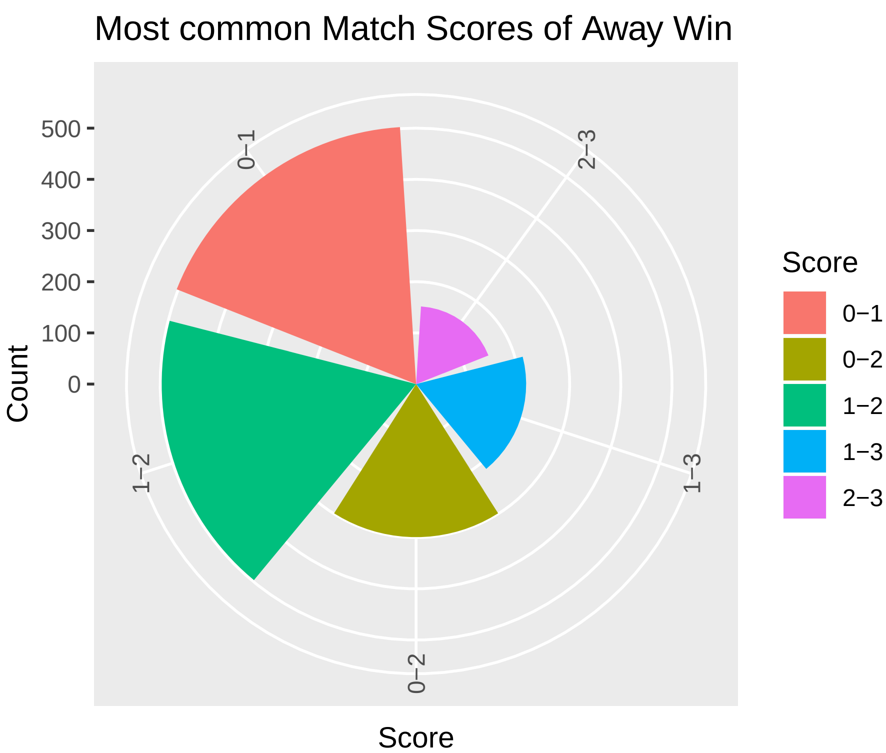
2.1. Home Win Most Common Match Scores

```
ggplot(common_home_win, aes(reorder(match_score, count), count, fill=match_score)) +geom_bar(stat="iden
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) + coord_polar() +
  labs(x="Score", y="Count", title="Most common Match Scores of Home Win", fill="Score")
```



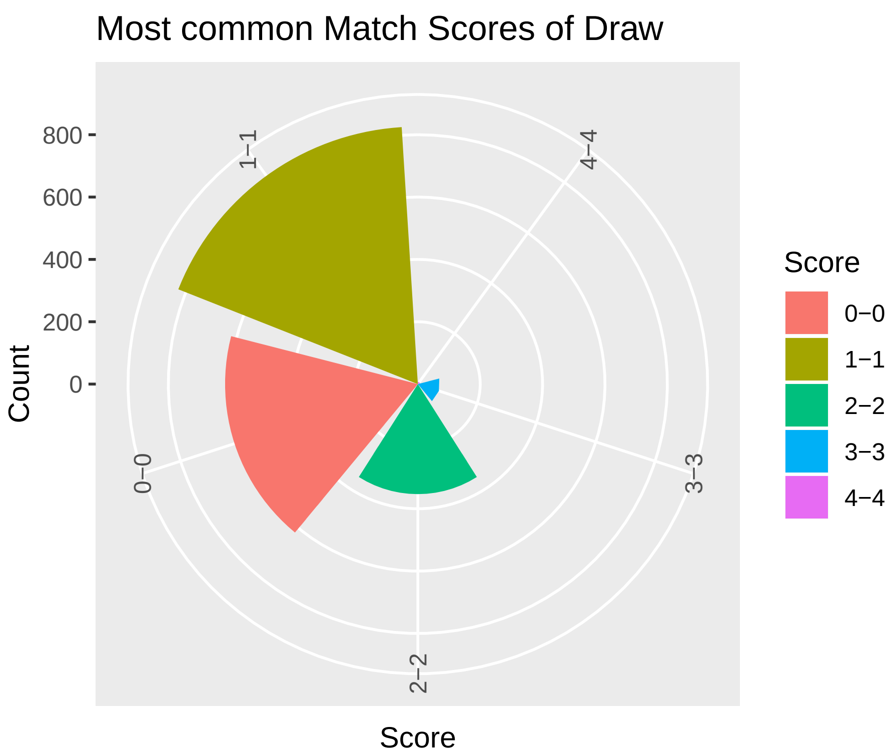
2.2. Away Win Most Common Match Scores

```
ggplot(common_away_win, aes(reorder(match_score, count), count, fill=match_score)) +geom_bar(stat="iden
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) + coord_polar() +
    labs(x="Score", y="Count", title="Most common Match Scores of Away Win", fill="Score")
```



2.3. Draw Most Common Match Scores

```
ggplot(common_draw, aes(reorder(match_score, count), count, fill=match_score)) +geom_bar(stat="identity
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) + coord_polar() +
    labs(x="Score", y="Count", title="Most common Match Scores of Draw", fill="Score")
```



3. Welcome to Real Life

3.1. Data Preparation

```
## total city values extracted into total_data
total_data <- all_data %>% filter(str_detect(sehir,"toplam") & !str_detect(sehir,"genel") & !str_detect
total_data$sehir <- gsub('toplam ', '', total_data$sehir)
total_data <- total_data %>% select(sehir, yil, sayi, uretim, toplama, nadas, toplam, miktar)

## city detail values extracted into all_data
all_data <- all_data %>% filter(!str_detect(sehir,"toplam")) %>% select(sehir, urun, miktar, yil)
glimpse(all_data)
```

```
## Observations: 12,957
## Variables: 4
## $ sehir  <chr> "adana", "adana", "adana", "adana", "adana", "adana", "adana",...
## $ urun   <chr> "acur", "ahududu", "alic(dogadan toplama)", "armut", "arpa", "...
## $ miktar <int> 200, 100, 40000, 79, 16483, 216495, 17, 32274, 125, 200, 1400,...
## $ yil    <int> 2014, 2014, 2014, 2014, 2014, 2014, 2014, 2014, 2014, 2014, 20...
```

```
glimpse(total_data)
```

```
## Observations: 386
## Variables: 8
## $ sehir   <chr> "adana", "adiyaman", "afyonkarahisar", "agri", "aksaray", "am...
```

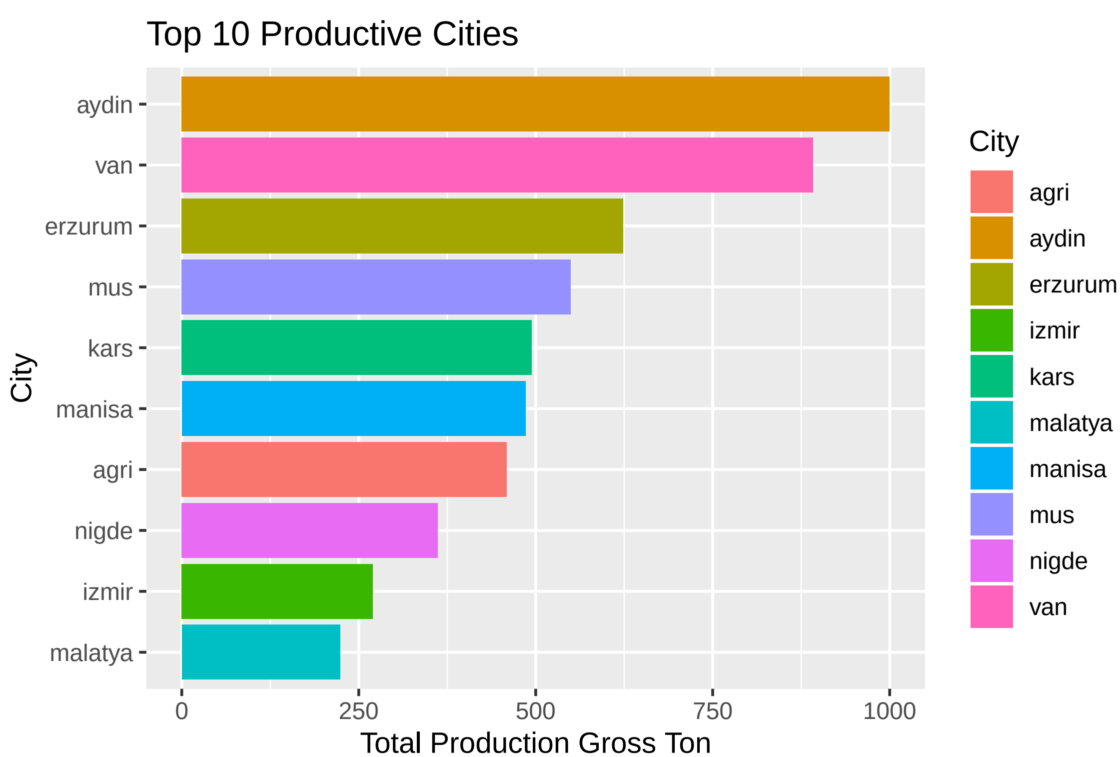
```
## $ yil     <int> 2014, 2014, 2014, 2014, 2014, 2014, 2014, 2014, 2014, 2014, 2...
## $ sayi    <dbl> 170, 94, 248, 1473, 1, 8, 34, 39, 275, 548, 4231, 71, 11, 13,...
## $ uretim  <dbl> 778.6883, 446.8956, 1058.9791, 28986.0259, 0.5550, 7.0792, 21...
## $ toplama <dbl> 1490.0000, 0.0000, 0.0000, 0.0000, 0.0000, 1200.0000, 0.0000,...
## $ nadas   <dbl> 0.2330, 0.0000, 29.3610, 415.7547, 0.0540, 0.0000, 15.1655, 0...
## $ toplam  <dbl> 2268.9213, 446.8956, 1088.3401, 29401.7806, 0.6090, 1207.0792...
## $ miktar  <int> 18699203, 2333020, 7777985, 85151588, 555, 131819, 8585544, 3...
```

3.2. Analyses

3.2.1 Gross Production of 10 Top Cities

```
gross_production <- all_data %>% filter(!is.na(miktar)) %>% group_by(sehir) %>%
  summarise(total_production = sum(miktar / 1000000)) %>%
  arrange(desc(total_production)) %>% head(10)

ggplot(gross_production, aes(reorder(sehir,total_production), total_production, fill=sehir))+
  geom_bar(stat='identity') + coord_flip() + labs(x="City", y="Total Production Gross Ton", fill="City"
        title="Top 10 Productive Cities")
```

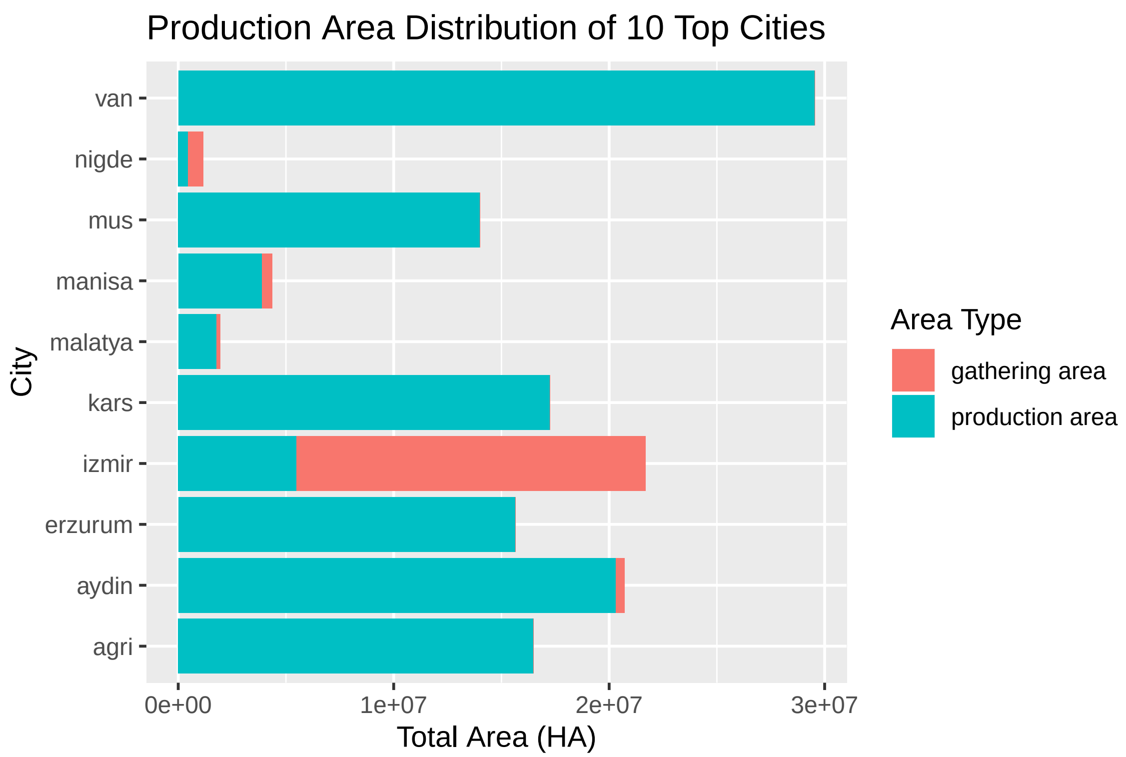


- Let's store these cities

```
top_cities <- as.vector(gross_production$sehir)
```

3.2.2. Distribution of Real Production / Gathering Area of 10 Top Cities

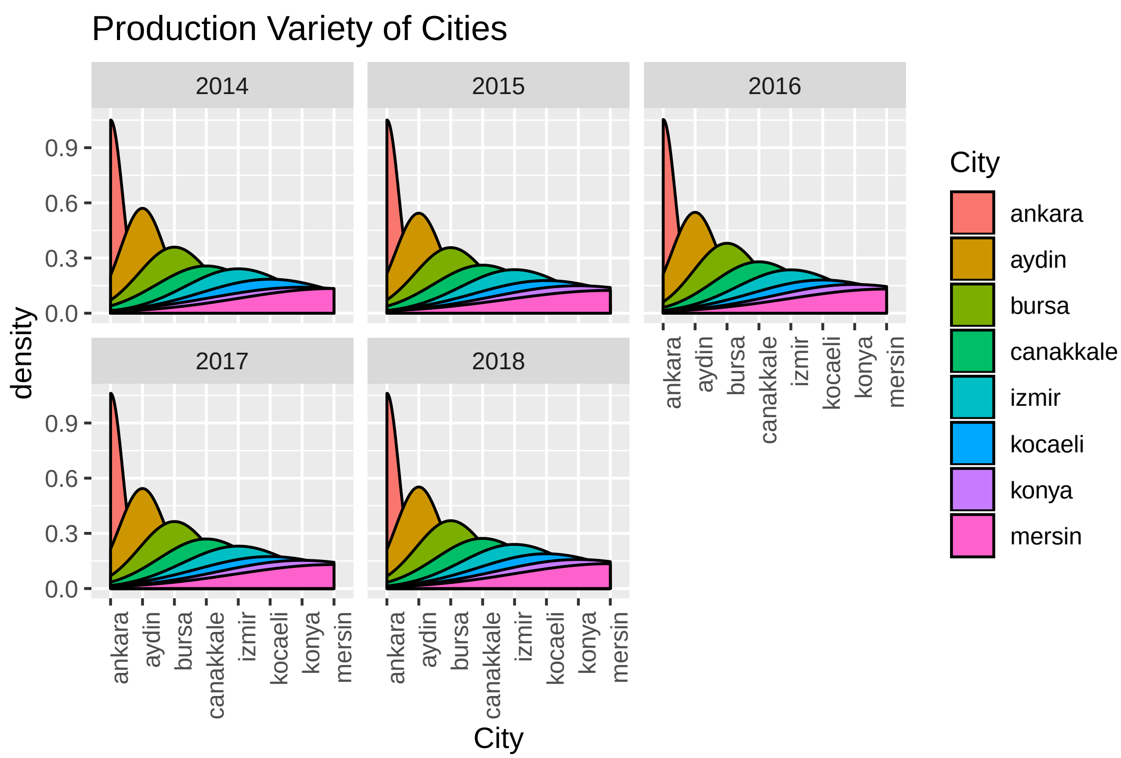
```
proportion_uretim <- total_data %>% filter(sehir %in% top_cities) %>% group_by(sehir) %>%
  summarise(total = (sum(uretim)) * 100,  type="production area")
proportion_toplama <- total_data %>% filter(sehir %in% top_cities) %>% group_by(sehir) %>%
  summarise(total = (sum(toplama)) * 100,  type="gathering area")
proportion_all = bind_rows(proportion_uretim, proportion_toplama)
ggplot(proportion_all, aes(sehir, total, fill=type)) + geom_bar(stat="identity", position="stack") + coo
  labs(x="City", y="Total Area (HA)", fill="Area Type", title="Production Area Distribution of 10 Top Ci
```



3.2.3 Poduction Variety of Top 5 Cities By Years

```
variety <- all_data %>% filter(miktar > 0) %>% group_by(yil, sehir) %>% summarise(count=n()) %>% top_n(
variety_cities <- as.vector(variety$sehir)
top_variety <- all_data %>% filter(sehir %in% variety_cities)

ggplot(top_variety, aes(sehir, fill=sehir)) + geom_density() + facet_wrap(~yil) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) + labs(x="City", fill="City", title="Producti
```



3.2.4 Most Nonfertile City Records

```
fertility <- total_data %>% filter(!is.na(nadas) & nadas > 0 & !is.na(toplam) & toplam > 0 ) %>%
  transmute(city = sehir,  percentage= (nadas/toplam) * 100, non_fertile_area=nadas, total_area=toplam,
  arrange(desc(non_fertile_area)) %>% head(10)
fertility
```

```
##       city percentage non_fertile_area total_area year
## 1      van   3.233483         2528.773  78205.878 2014
## 2      van   3.298937         1970.509  59731.650 2015
## 3  erzurum   4.500435         1777.608  39498.572 2015
## 4  erzurum   4.652970         1501.536  32270.475 2016
## 5      van   2.474571         1438.506  58131.553 2016
## 6  erzurum   4.192321         1376.269  32828.325 2014
## 7      van   1.932645         1094.850  56650.348 2017
## 8    sivas  12.223955         1006.397   8232.990 2015
## 9    sivas   9.095826          783.837   8617.546 2014
## 10 erzurum   2.692013          768.095  28532.362 2018
```