North Korea Missile Test Example Data Analysis

adenoz

2022-11-27

Introduction

This document aims to provide an example of data analysis that is conducted against an *event* dataset. This event dataset contains curated and carefully structured data on North Korean missile tests.

This analysis was conducted using the R programming language, in an R Markdown notebook using the RStudio Integrated Development Environment (IDE). While this analysis was conducted in R, it could just as easily have been conducted in Python or Julia is Jupyter Notebooks.

Analytical notebooks allow analysts to write prose, like this introduction, while also including *code blocks* that conduct some action against the underlying dataset that is imported. Typically this can include data ingestion, wrangling (cleaning, ensuring consistent / correct data and data types, dealing with erroneous or incorrect data etc), tidying (into formats / structures suitable for analysis), transformations (changing scales, generating new variables, joining with other data etc) and importantly - visualisation and modelling / forecasting.

The PDF document you are reading is an example of what a notebook (whether R Markdown or Jupyter Notebook) can be exported as. The actual code blocks can all be included in the output, a selection can be included, or none can be included. This will depend on your audience. If you are sharing an analysis with other technical users, you will probably want to include the code so they can see what has been done. For a non technical reader, no code can be included. This means the report will consist of written prose along with the graphs and plots and tables and other outputs produced from the actual underlying code. The written prose should be relating to the output of the code blocks, and not the actual code at all.

In this example, most though not all of the code used will be included as the intent of this is to expose analysts to what code is used to generate the various outputs. However just be aware that if the purpose was to communicate knowledge about data, none of the actual code is needed to explain the analysis and logic and conclusions arrived at. Note that in the code blocks, lines beginning with the # are known as comments. This is not actually code, but provides some explanations of what is happening in the code. The program does not read any comments, the program skips any line beginning with #, they are purely there for human readers.

And finally before we get into it, this document shows the progress and thinking and exploring as we get deeper and deeper into a dataset. This analysis is known as *exploratory data analysis*. In this form, it may not all be required to be in a final report. Typically, you would go through this exploratory phase, find out interesting things, then produce a separate more focused more concise report that doesn't take a reader on a journey, but just shows them the so what. However this is an example aimed at analysts, so it does show the exploratory journey.

The data

This dataset is maintained and provided by the James Martin Center for Nonproliferation Studies (CNS). It was downloaded from this site.

This dataset contains flight tests of all missiles launched by North Korea capable of delivering a payload of at least 500kg a distance of at least 300km. The first such test in this dataset is from April 1984 and the most recent as at the time of this analysis being October 2022. Updates to this dataset are normally conducted in as short as two or three days after a test.

The dataset consists of two csv files. One contains details of the actual missile tests, the other contains additional details of all of the launch facilities. In this analysis, we'll be joining both tables so that we can use data from both tables, however it makes sense to maintain different tables as they are logically about different though related things. To join tables, we simply need to identify a single column in each table that contains the exact same information. These identical columns form the key in which to join the two tables.

Note right away upon import that we need to do some minor cleaning and tidying of the data. This is a huge part of conducting data analysis. Note also that some lines of what looks like code has been 'commented out'. This means that during import these lines were used to check various parts of the data, but we didn't need to include them in a report. However rather than delete them, we simply 'comment them out' so that it is easy to just delete the # to use those lines again in future analyses.

```
# Import the two csv files
fac.df = read.csv("data_copy/Facilities.csv")
mis.df = read.csv("data_copy/Missile_Tests.csv")
# the header = TRUE does not work properly
#mis.tt = mis.df  # for testing and experimentation purposes
# copy first row to be column names, then delete that top row
names(fac.df) = as.character(fac.df[1,])
fac.df = fac.df[-1,]
# copy first row to be column names, then delete that top row
names(mis.df) = as.character(mis.df[1,])
mis.df = mis.df[-1,]
# explore and make sure import worked correctly
#View(mis.df)  # View opens the data in a new tab to visually explore.
#tail(mis.df)
```

Research questions

Firstly, we are probably not going to crack the nut of the North Korean missile testing program here. However we can probably build a pretty good understanding of the North Korean missile testing program through our analysis of this dataset. I personally had zero expertise of the North Korean missile testing program beyond what comes up in the media from time to time but built a pretty good understanding quite quickly. Analysing data, if the data is good, can provide moderate understanding far quicker than reading large quantities of qualitative reports from subject matter experts. Let's see what we can learn.

We hope through our analysis to be able to answer the following questions. Perhaps other questions will arise as we explore and understand the dataset more, but these will get us going.

- is North Korea's missile testing program becoming higher tempo?
- is North Korea's missile testing program becoming more successful?
- are the facilities being used to launch the most important missiles changing over time?
- what are North Korea's most important missiles?
- what are North Korea's most capable missiles (if different to above)?
- what are the preferred launch times of day?
- what are the preferred launch days, if any?
- what are the preferred landing locations?
- what are the most important missile launch facilities?
- what are the most used launch facilities?

- what launch facilities are only, or mostly, used for the longest range missiles?

We'll potentially come across unexpected interesting findings as we explore the data.

Data cleaning

We'll need to some minor data cleaning, mainly around data types.

```
#View(mis.df)
#mis.tt = mis.df

# DATES=====================================
# Dates are not all equal format.
# for the half dozen dates without days, we'll simply add the 1st. not many, should be fine.
mis.df[4:6,2] = "1-Sep-84"
mis.df[7,2] = "1-May-86"
mis.df[8,2] = "1-May-90"
mis.df[9,2] = "1-Jun-90"
mis.df[10,2] = "1-Jul-91"
mis.df[11,2] = "1-Jun-92"

# the following will only do the months with day - month - year format.
mis.df$Date = as.Date(mis.df$Date, format = "%d-%b-%y")
# same for date entered variable / column
#mis.tt$`Date Entered/Updated` = as.Date(mis.tt$`Date Entered/Updated`, format = "%d-%b-%y")

# copy unchanged date so we don't lose dates with no times
mis.df = mis.df %>%
  mutate(date_launch = Date)

# TIME======================================
# using tidyr for separate
mis.df = mis.df %>%
  separate(`Launch Time (UTC)`, c('launched', 'when'), sep=" ")

mis.df$dateandtime <- as.character(paste(mis.df$Date, mis.df$launched, sep = ' '))

mis.df$dateandtime = as.POSIXct(mis.df$dateandtime, format = "%Y-%m-%d %H:%M:%S")

hourz = 12 * 60 * 60

mis.df$dateandtime[mis.df$when == "pm" & !is.na(mis.df$dateandtime)] =
  mis.df$dateandtime[mis.df$when == "pm" & !is.na(mis.df$dateandtime)] + hourz

# UTC to North Korea (Pyonyang) time
# Note the original data is in UTC, need to add 9hrs

utc_nk = 9 * 60 * 60

mis.df$dateandtime[!is.na(mis.df$dateandtime)] =
  mis.df$dateandtime[!is.na(mis.df$dateandtime)] + utc_nk
```

```r
# move new variables closer to front for easy checking
mis.df = mis.df %>%
  select(F1, date_launch, dateandtime, everything())

#mis.tt$launched = as.POSIXct(mis.tt$launched, format = "%H:%M:%S")  # different dates

# Apogee and distance travelled====================
# NA values. disregard, I think NAs will be ok. can filter out
#mis.tt$Apogee[is.na(mis.tt$Apogee)] = "Unknown"
#mis.tt$`Distance Travelled`[is.na(mis.tt$`Distance Travelled`)] = "Unknown"

# Will probably want to change Unknown to NA so we can change the variable type to Int
mis.df$Apogee[mis.df$Apogee == "Unknown"] = NA
mis.df$Apogee[mis.df$Apogee == "N/A"] = NA
mis.df$`Distance Travelled`[mis.df$`Distance Travelled` == "Unknown"] = NA
mis.df$`Distance Travelled`[mis.df$`Distance Travelled` == "N/A"] = NA

# There are some values with ranges. We'll edit them manually to be the centre.
# only two observations
mis.df$Apogee[mis.df$Apogee == "between 25 and 90 km"] = "57"
mis.df$`Distance Travelled`[mis.df$`Distance Travelled` == "between 110 and 670 km"] = "390"
# and one with a comma which needs fixing
mis.df$`Distance Travelled`[mis.df$`Distance Travelled` == "1,380 km"] = "1380 km"

# remove the 'km' from the values
mis.df = mis.df %>%
  separate(Apogee, c('height', 'km'), sep=" ")
mis.df = mis.df %>%
  separate(`Distance Travelled`, c('distance', 'kmz'), sep=" ")
mis.df$height = as.integer(mis.df$height)
mis.df$distance = as.integer(mis.df$distance)
# ^ note, we removed the decimal point on a couple of locations, which should be ok

# remove no longer needed variables.
# confirm created variables are correct first, then do this final select comd
mis.df = mis.df %>%
 select(-c(F1, Date, `Date Entered/Updated`, launched, km, kmz, 'NA'))

# tail(mis.df, 20)
```

Graphical data exploration

Timeseries plot of all tests

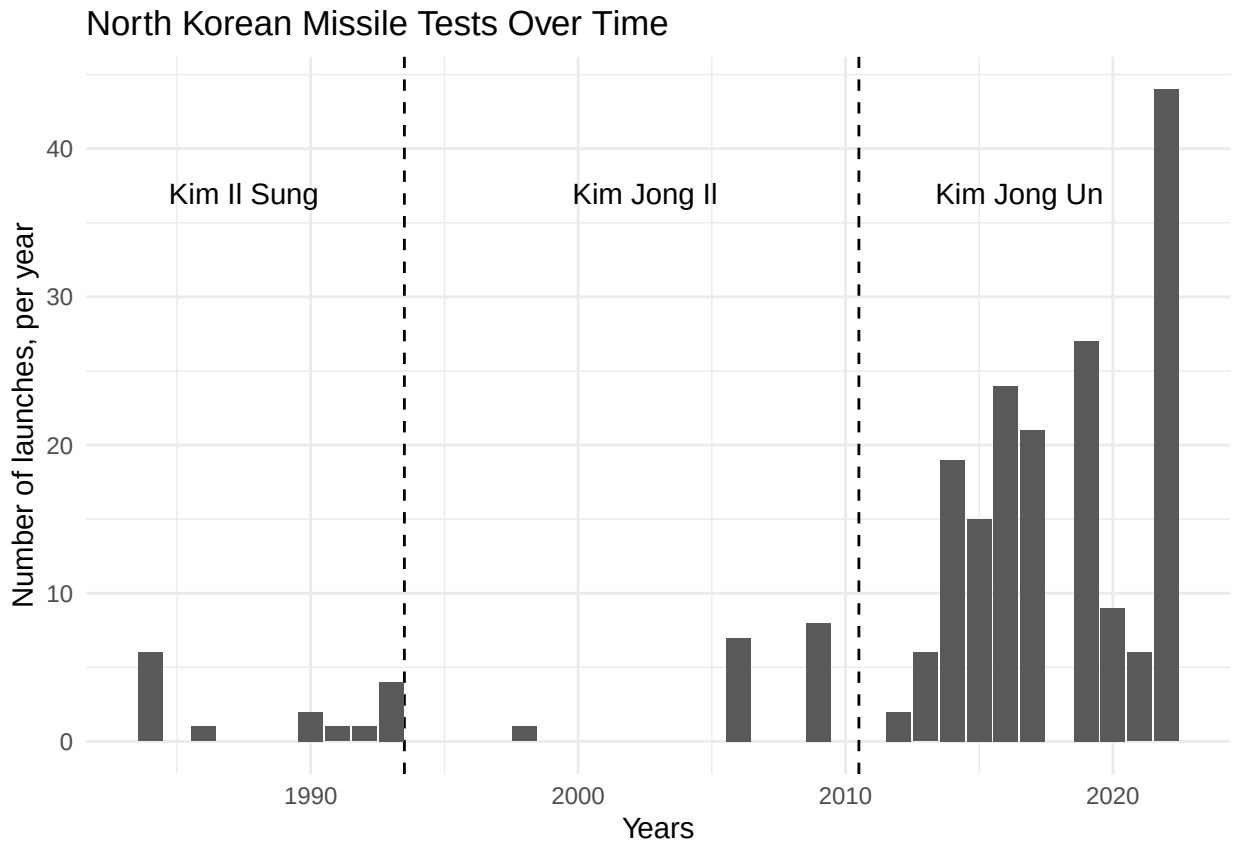
To begin, it probably makes the most sense to plot all launches along a timeseries plot to see how the overall number of monthly launches have evolved over time. This should give us a good overview of the data at a basic high level. We'll also annotate the times of the three generations of leaders in order to compare any trends associated with them. The following plot groups the data by year, and aggregates by simply counting the events within each year. Note that there are some years with zero counts. When this happens, the line simply connects directly to the next year with some value. A quick comment on this.. typically plots that show data over time use a line graph. This is best practice as it allows us to plot a number of variables at the same time keeping things clean and clutter free. However for data that is 'count' data when there are

zero values, like this data, it doesn't really show these zero values well. But in this case it does clearly show changes in trends.

```
# note floor_date uses lubridate
mis.df %>%
  group_by(year=floor_date(date_launch, "year")) %>%
  summarise(count = n()) %>%
  ggplot(aes(year, count)) +
  #ylim(0,15)+
  # the following years are one year earlier than actual date due to way plot renders
  geom_vline(aes(xintercept = as.Date("1993", format = "%Y")), lty = 2)+
  #geom_rect(aes(xmin=year[1], xmax=year[6], ymin=0, ymax=Inf, fill='yellow'))+
  geom_vline(aes(xintercept = as.Date("2010", format = "%Y")), lty = 2)+
  geom_line(linewidth = 0.7, col = 'red')+
  #geom_text("hello", )
  annotate("text", x = as.Date("1987", format = "%Y"), y = 37, label =  "Kim Il Sung")+
  annotate("text", x = as.Date("2002", format = "%Y"), y = 37, label =  "Kim Jong Il")+
  annotate("text", x = as.Date("2016", format = "%Y"), y = 37, label =  "Kim Jong Un")+
  labs(title = "North Korean Missile Tests Over Time",
       subtitle = "Non-zero values connect directly in absense of launches",
       x = "Years",
       y = "Number of launches, per year")+
  theme_minimal()
```

We'll now present the same data, but use bars instead of a line.

```r
# note floor_date uses lubridate
mis.df %>%
  group_by(year=floor_date(date_launch, "year")) %>%
  summarise(count = n()) %>%
  ggplot(aes(year, count)) +
  geom_vline(aes(xintercept = as.Date("1993", format = "%Y")), lty = 2)+
  geom_vline(aes(xintercept = as.Date("2010", format = "%Y")), lty = 2)+
  geom_bar(stat = 'identity')+
  annotate("text", x = as.Date("1987", format = "%Y"), y = 37, label =  "Kim Il Sung")+
  annotate("text", x = as.Date("2002", format = "%Y"), y = 37, label =  "Kim Jong Il")+
  annotate("text", x = as.Date("2016", format = "%Y"), y = 37, label =  "Kim Jong Un")+
  labs(title = "North Korean Missile Tests Over Time",
       x = "Years",
       y = "Number of launches, per year")+
  theme_minimal()
```



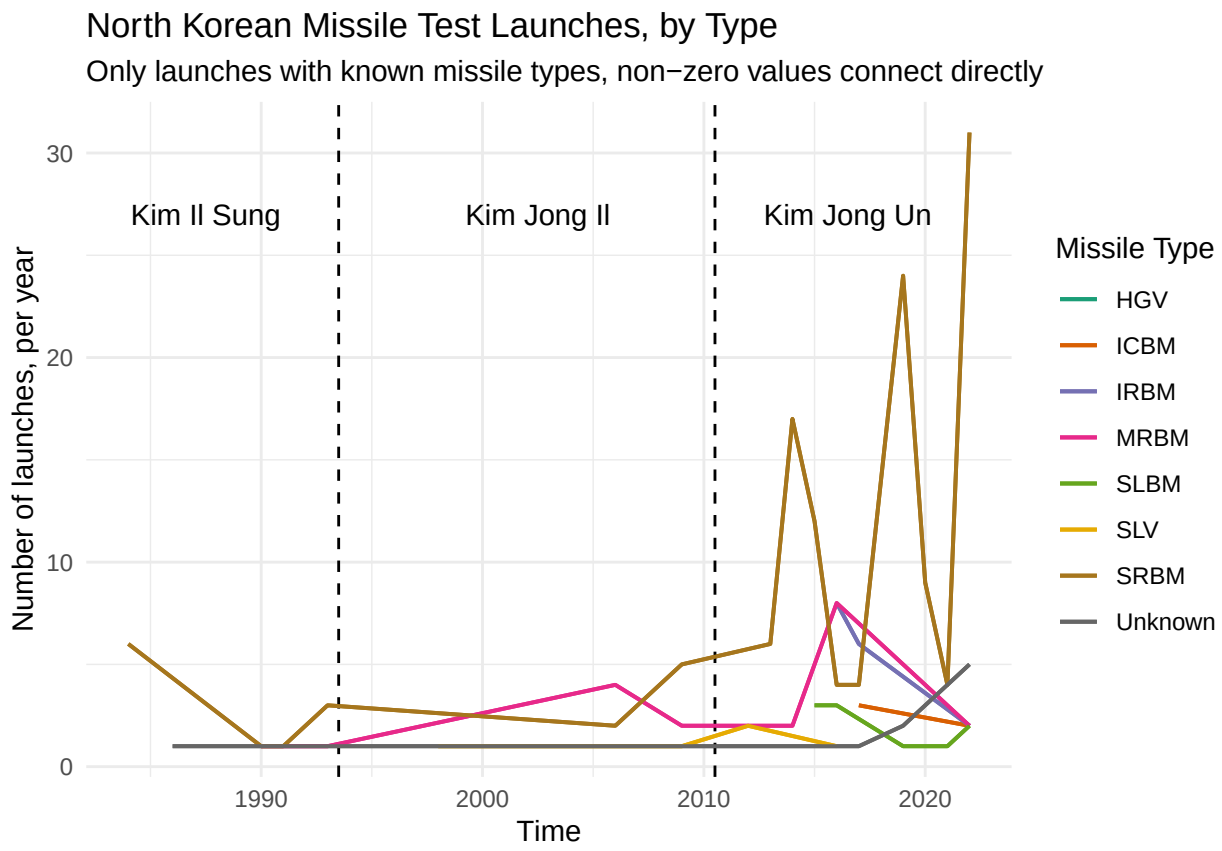
Because this plot is essentially counts and there are many years with values of zero, it makes more sense to use bars instead of a line. This is an exception to the rule, that data over time typically plots better using lines.

Regardless, we can see a significant uptick in missile tests from around 2013 and onwards during the leadership of Kim Jong Un.

Launches over time, by missile type

Let's now break that down by missile type to see if there has been a change in the type of missile being tested. While the previous in mind regarding lines vs bars, we'll now go back to using lines as we have multiple data points. Stacks bars are much less intuitive and are harder to interpret than multiple lines. Noting the issues around zero values, we'll use lines in the following as we're mainly interested in comparing values with each other, not so much interested in zero values at this point.

```
mis.df %>%
  group_by(year=floor_date(date_launch, "year"), `Missile Type`) %>%
  summarise(count = n()) %>%
  ggplot(aes(year, count, col = `Missile Type`)) +
  geom_line(linewidth = 0.7)+
  # the following years are one year earlier than actual date due to way plot renders
  geom_vline(aes(xintercept = as.Date("1993", format = "%Y")), lty = 2)+
  #geom_rect(aes(xmin=year[1], xmax=year[6], ymin=0, ymax=Inf, fill='yellow'))+
  geom_vline(aes(xintercept = as.Date("2010", format = "%Y")), lty = 2)+
  geom_line(linewidth = 0.7)+
  #geom_text("hello", )
  annotate("text", x = as.Date("1987", format = "%Y"), y = 27, label =  "Kim Il Sung")+
  annotate("text", x = as.Date("2002", format = "%Y"), y = 27, label =  "Kim Jong Il")+
  annotate("text", x = as.Date("2016", format = "%Y"), y = 27, label =  "Kim Jong Un")+
  scale_color_brewer(palette = "Dark2") +
  labs(title = "North Korean Missile Test Launches, by Type",
       subtitle = "Only launches with known missile types, non-zero values connect directly",
       x = "Time",
       y = "Number of launches, per year")+
  theme_minimal()
```



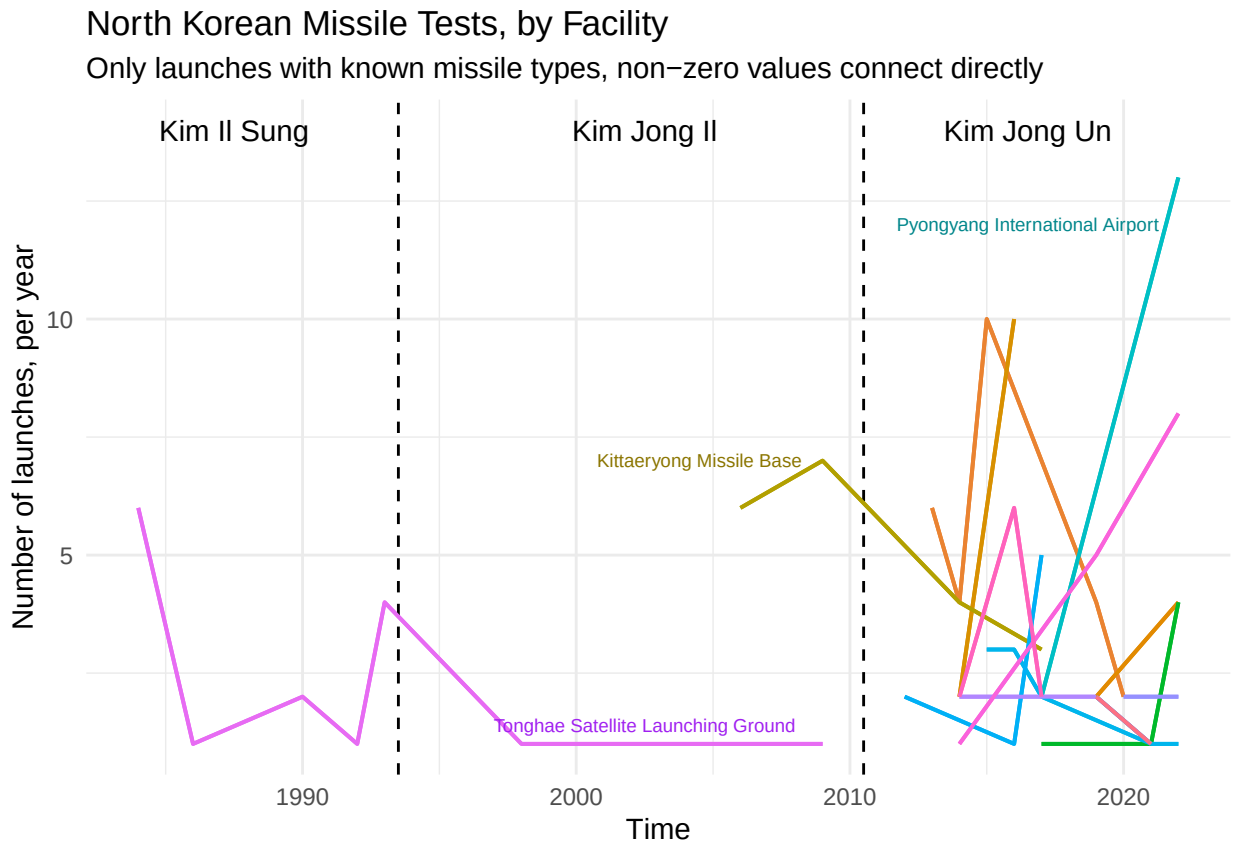
By far, the most commonly used missile to test under Kim Jong Un has been the Short Range Ballistic Missile (SRBM). Interestingly, there is an increasing number of missile tests where the type is being recorded as *Unknown*. Un has also tested Inter-Continental Ballistic Missiles (ICBM), Intermediate-Range Ballistic Missiles (IRBM) and Submarine-Launched Ballistic Missiles (SLBM). These are all concerning offensive capabilities and the escalation by Un is clear.

Use of facilities over time

We'll look at another timeseries plot, this time comparing the use of launch facilities.

```
mis.df %>%
  group_by(year=floor_date(date_launch, "year"), `Facility Name`) %>%
  summarise(count_fac = n()) %>%
  #filter(count_fac > 0) %>%
  ggplot(aes(year, count_fac, col = `Facility Name`)) +
  geom_line(linewidth = 0.7)+
  # the following years are one year earlier than actual date due to way plot renders
  geom_vline(aes(xintercept = as.Date("1993", format = "%Y")), lty = 2)+
  #geom_rect(aes(xmin=year[1], xmax=year[6], ymin=0, ymax=Inf, fill='yellow'))+
  geom_vline(aes(xintercept = as.Date("2010", format = "%Y")), lty = 2)+
  geom_line(linewidth = 0.7)+
  #geom_text("hello", )
  annotate("text", x = as.Date("1987", format = "%Y"), y = 14, label =  "Kim Il Sung")+
  annotate("text", x = as.Date("2002", format = "%Y"), y = 14, label =  "Kim Jong Il")+
  annotate("text", x = as.Date("2016", format = "%Y"), y = 14, label =  "Kim Jong Un")+
```

```
  annotate("text", x = as.Date("2002", format = "%Y"), y = 1.4, label =  "Tonghae Satellite Launching G
  annotate("text", x = as.Date("2004", format = "%Y"), y = 7, label =  "Kittaeryong Missile Base", cex =
  annotate("text", x = as.Date("2016", format = "%Y"), y = 12, label =  "Pyongyang International Airpor
  #scale_color_brewer(palette = "Dark2") +
  labs(title = "North Korean Missile Tests, by Facility",
       subtitle = "Only launches with known missile types, non-zero values connect directly",
       x = "Time",
       y = "Number of launches, per year")+
  theme_minimal()+
  theme(legend.position = "none")
```

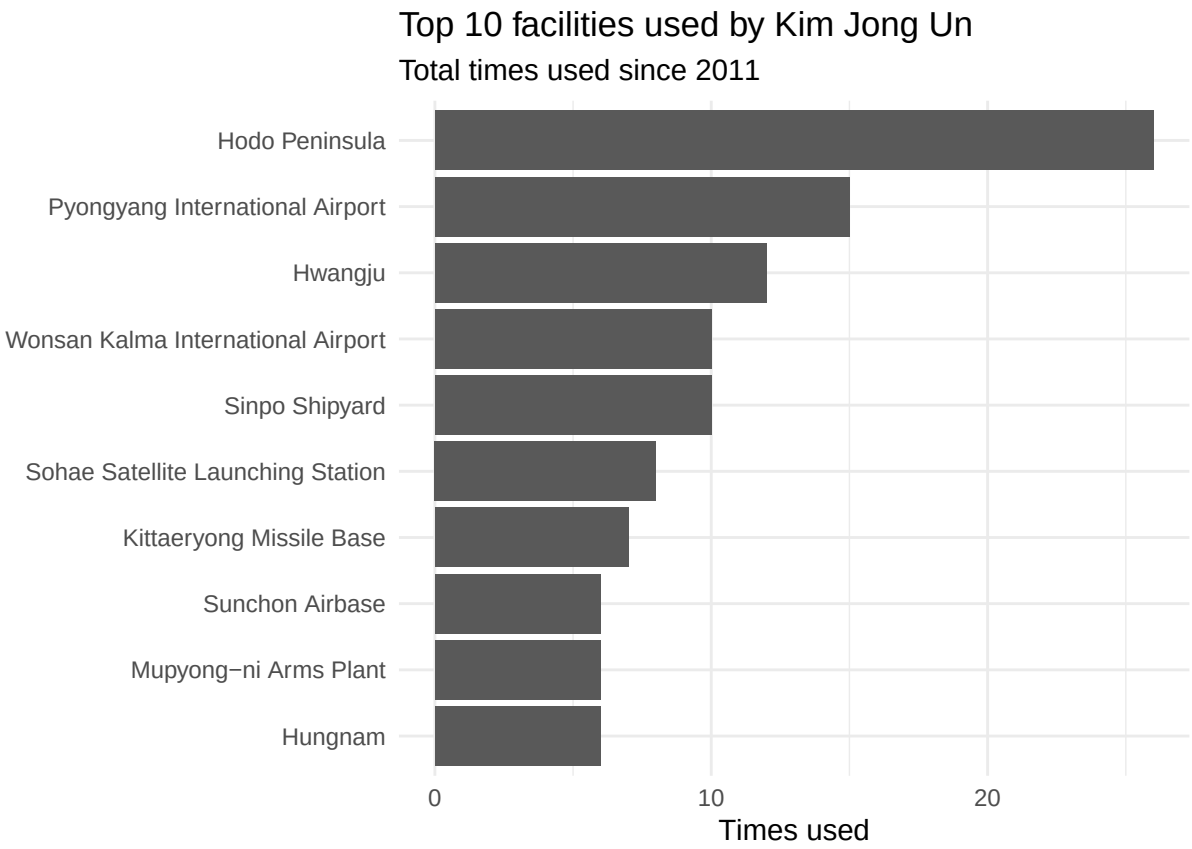


We've hidden the legend for the above plot as that makes things quite busy. But regardless, we can see that Kim Il Sung mostly used one launch facility, being the Tonghae Satellite Launching Ground. We can see that Kim Jong Il continued using that facility for some time then started using a different facility, being the Kittaeryong Missile Base. But then with Kim Jong Un, he has dramatically changed and varied the facilities used to conduct missile tests. Un's most used facility in recent years has been at Pyongyang International Airport.

Facilities used by Kim Jong Un

Let's look at the top ten facilities used during the leadership of Un. The following plot shows the count of missile tests conducted at the various facilities since 2011, sorted, in order.

```r
mis.df %>%
  filter(date_launch > "2011-01-01" & `Facility Name` != "Unknown") %>%
  count(`Facility Name`) %>%
  arrange(desc(n)) %>%
  head(10) %>%
  ggplot(aes(reorder(`Facility Name`, n), n))+
  geom_bar(stat = 'identity')+
  coord_flip()+
  labs(title = "Top 10 facilities used by Kim Jong Un",
       subtitle = "Total times used since 2011",
       x = "",
       y = "Times used")+
  theme_minimal()
```

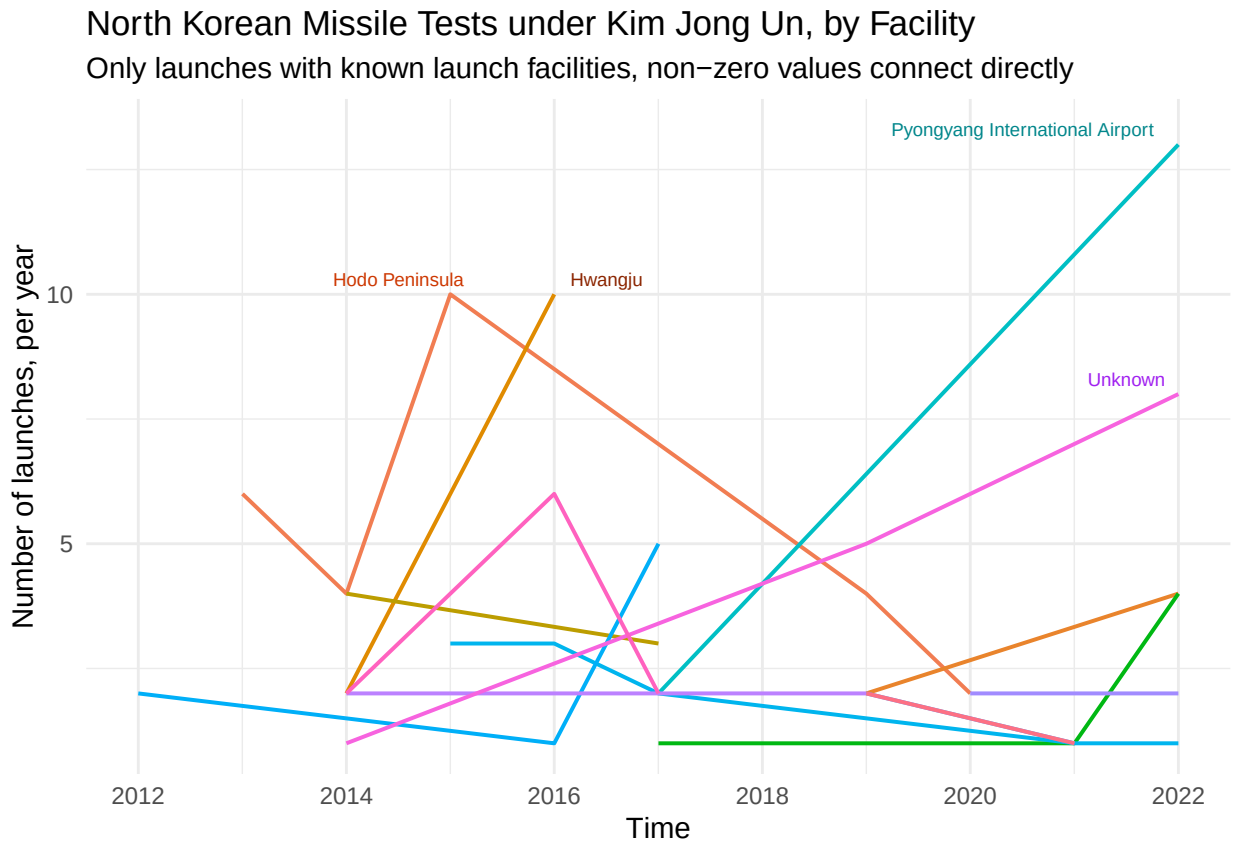


Facilities used by Kim Jong Un over time

We'll now look at the facilities used by year to see how Un's use of the facilities changed over time. There are many different facilities, so using stacked bars is very messy and the legend becomes just as large as the graphical plot. With the line plot, it's still rather messy, though we can highlight just those with the highest counts and ignore labelling the others.

```r
mis.df %>%
  filter(date_launch > "2011-01-01") %>%
  group_by(year=floor_date(date_launch, "year"), `Facility Name`) %>%
```

```
  summarise(count_fac = n()) %>%
  #filter(count_fac > 0) %>%
  ggplot(aes(year, count_fac, col = `Facility Name`)) +
  geom_line(linewidth = 0.7)+
  annotate("text", x = as.Date("2014", format = "%Y"), y = 10.3, label =  "Hodo Peninsula", cex = 2.4,
  annotate("text", x = as.Date("2021", format = "%Y"), y = 8.3, label =  "Unknown", cex = 2.4, color =
  annotate("text", x = as.Date("2020", format = "%Y"), y = 13.3, label =  "Pyongyang International Airp
    annotate("text", x = as.Date("2016", format = "%Y"), y = 10.3, label =  "Hwangju", cex = 2.4, color
  #scale_color_brewer(palette = "Dark2") +
  labs(title = "North Korean Missile Tests under Kim Jong Un, by Facility",
       subtitle = "Only launches with known launch facilities, non-zero values connect directly",
       x = "Time",
       y = "Number of launches, per year")+
  theme_minimal()+
  theme(legend.position = "none")
```



We can see that Pyongyang International Airport has been use the most in the most recent data. Let's now look at that facility more closely.

Pyongyand International Airport

Let's see what missiles are being tested at Pyongyang International Airport to get a sense of how the facility is being used. As well as plots, we can arrange neatly formatted tables.

```
kable(mis.df %>%
  filter(date_launch > "2011-01-01" & `Facility Name` == "Pyongyang International Airport") %>%
  group_by(year=floor_date(date_launch, "year"), `Missile Type`) %>%
  summarise(count_type = n()) %>%
  arrange(year))
```

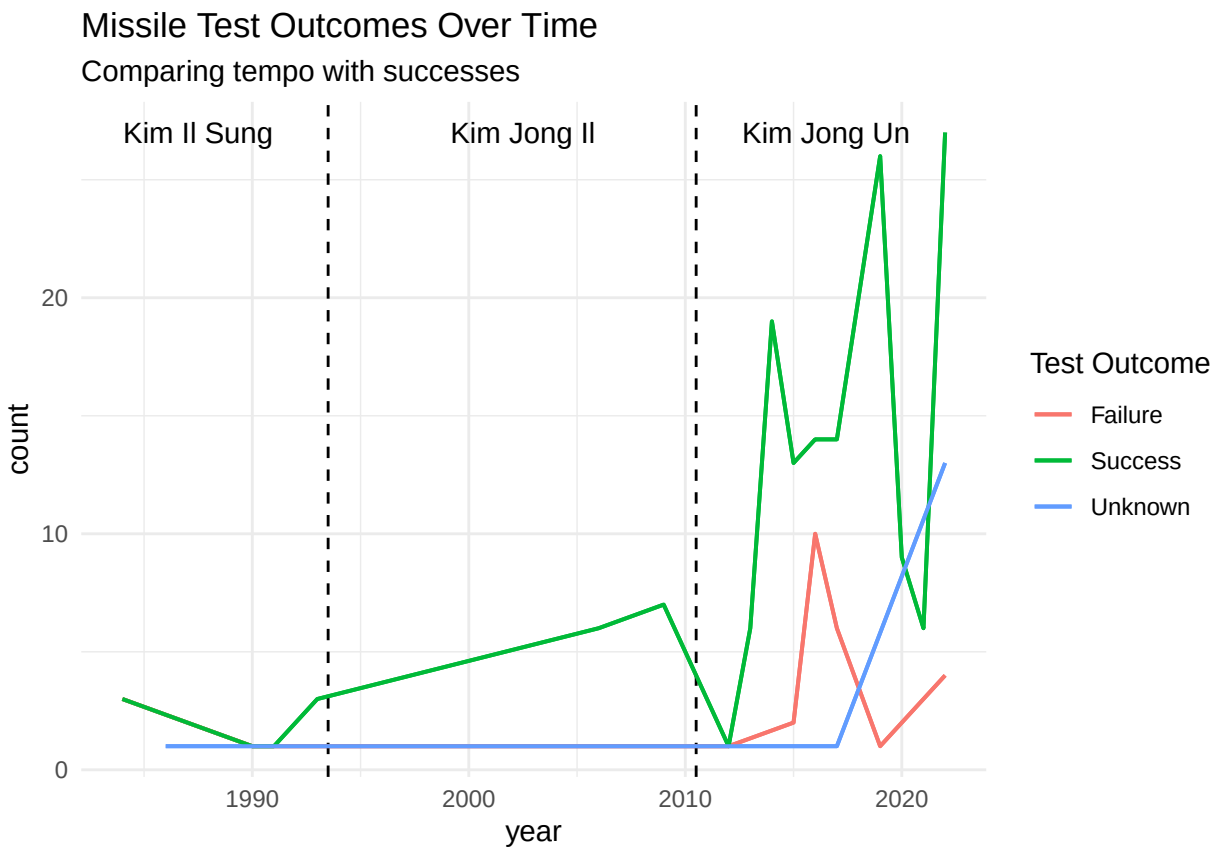
year	Missile Type	count_type
2017-01-01	IRBM	2
2022-01-01	ICBM	2
2022-01-01	SRBM	6
2022-01-01	Unknown	5

```
  # or plot, table probably looks better and more suitable in this case
  #ggplot(aes(year, count_type, col = `Missile Type`)) +
  #geom_bar(stat = 'identity')
```

Success and failures of missile tests

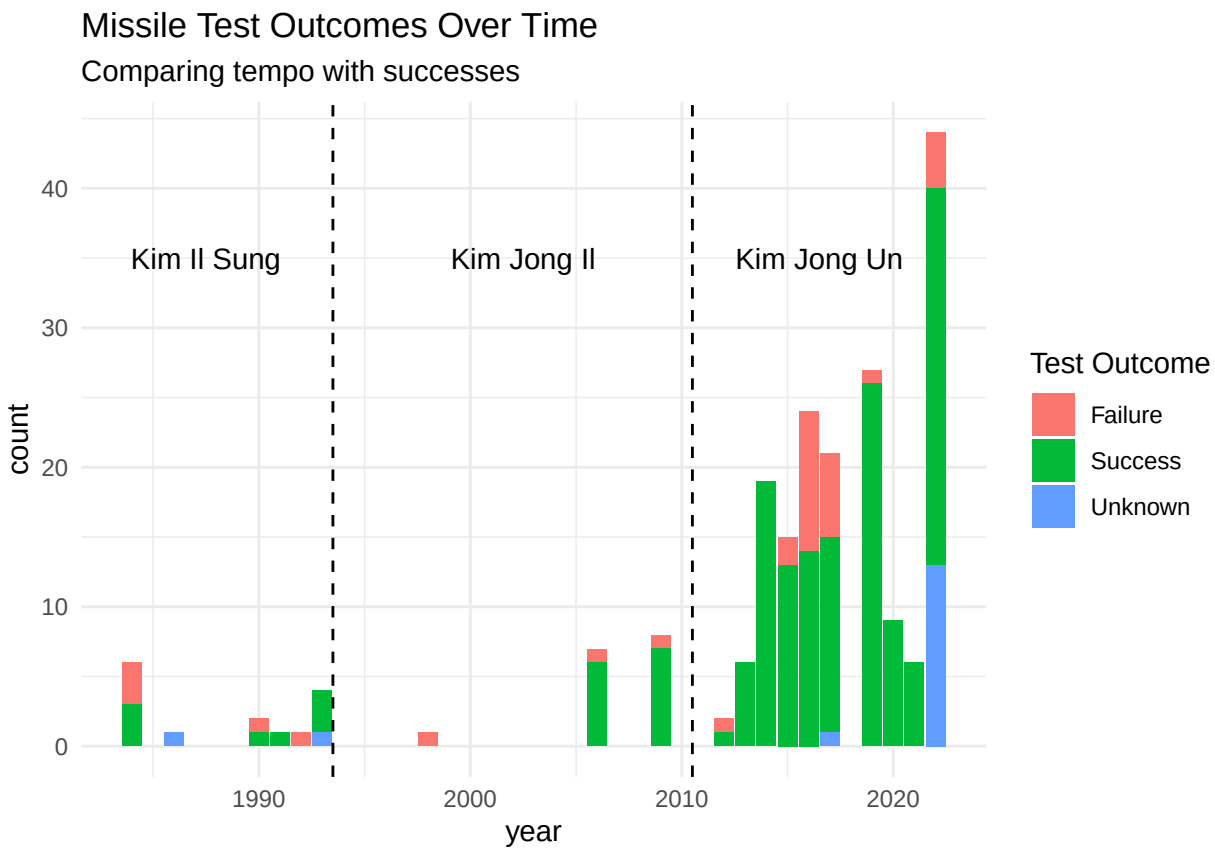
Let's now see if the dramatically increased quantity of missiles tests and the seemingly deliberate variation in facilities used has resulted in more successful tests.

```
mis.df %>%
  group_by(year=floor_date(date_launch, "year"), `Test Outcome`) %>%
  summarise(count = n()) %>%
  ggplot(aes(year, count, col = `Test Outcome`)) +
  geom_line(linewidth = 0.7)+
    # the following years are one year earlier than actual date due to way plot renders
  geom_vline(aes(xintercept = as.Date("1993", format = "%Y")), lty = 2)+
  #geom_rect(aes(xmin=year[1], xmax=year[6], ymin=0, ymax=Inf, fill='yellow'))+
  geom_vline(aes(xintercept = as.Date("2010", format = "%Y")), lty = 2)+
  geom_line(linewidth = 0.7)+
  #geom_text("hello", )
  annotate("text", x = as.Date("1987", format = "%Y"), y = 27, label =  "Kim Il Sung")+
  annotate("text", x = as.Date("2002", format = "%Y"), y = 27, label =  "Kim Jong Il")+
  annotate("text", x = as.Date("2016", format = "%Y"), y = 27, label =  "Kim Jong Un")+
  labs(title = "Missile Test Outcomes Over Time",
       subtitle = "Comparing tempo with successes")+
  theme_minimal()
```



Because the data in the above is less cluttered, we'll also use a stacked bar plot drawing on the exact same data.

```
mis.df %>%
  group_by(year=floor_date(date_launch, "year"), `Test Outcome`) %>%
  summarise(count = n()) %>%
  ggplot(aes(year, count, fill = `Test Outcome`)) +
  geom_bar(stat = 'identity')+
    # the following years are one year earlier than actual date due to way plot renders
  geom_vline(aes(xintercept = as.Date("1993", format = "%Y")), lty = 2)+
  #geom_rect(aes(xmin=year[1], xmax=year[6], ymin=0, ymax=Inf, fill='yellow'))+
  geom_vline(aes(xintercept = as.Date("2010", format = "%Y")), lty = 2)+
  #geom_line(linewidth = 0.7)+
  #geom_text("hello", )
  annotate("text", x = as.Date("1987", format = "%Y"), y = 35, label =  "Kim Il Sung")+
  annotate("text", x = as.Date("2002", format = "%Y"), y = 35, label =  "Kim Jong Il")+
  annotate("text", x = as.Date("2016", format = "%Y"), y = 35, label =  "Kim Jong Un")+
  labs(title = "Missile Test Outcomes Over Time",
       subtitle = "Comparing tempo with successes")+
  theme_minimal()
```



And in this case, one could make a case that the stacked bar plot is a better fit here. We not only get a sense of the zero years, but the volume or green paints a picture as well.

Missile type failures

Let's now see which missiles are the ones that are failing the most, by proportion tested. In the following code, we are doing a couple of simple calculations and arriving at a failure rate. The failure rate wasn't included in the original data, but we are able to compute it relatively easily. Such a rate or ratio can be a good way to rank or sort things. In this case, we also sort the table by the failure rate, in descending order.

```
# find failures
count_fails = mis.df %>%
  filter(`Test Outcome` == "Failure") %>%
  count(`Missile Type`)
  #head()
# count total tests
count_total = mis.df %>%
  count(`Missile Type`)
# inner join onto the failed data
counts = count_fails %>%
  inner_join(count_total, by = "Missile Type")
# rename new count columns
counts = rename(counts, Failed = n.x, Tests = n.y)
# calculate failure rate
counts = counts %>%
```

```
  mutate(`Fail Rate` = round((Failed / Tests ) * 100, 2))

kable(counts %>%
  arrange(desc(`Fail Rate`)))
```

Missile Type	Failed	Tests	Fail Rate
SLV	4	6	66.67
IRBM	10	16	62.50
Unknown	4	9	44.44
SLBM	3	10	30.00
ICBM	1	5	20.00
MRBM	5	28	17.86
SRBM	5	129	3.88

Note the above only lists those missiles with failures recorded. Very few SRBM tests have failed, and this is from the largest quantity of missile types tested. It would therefore be reasonable to judge that North Korea has a competent SRBM capability.

However it appears evident that they do not yet have a SLV capability, as over half of all tests have failed. The same can be said of their IRBM capability, and they have tested over double the amount compared to SLVs. SLVs, along with ICBMs, are amongst the most concerning long range missile systems. We'll look at range shortly.

Missile failures

Let's now dive in deeper and look at the actual missiles that are failing to see if we can learn more. We'll again perform some simple calculations but use the missile names and include the missile type, sorted by fail rate.

```
ncount_fails = mis.df %>%
  filter(`Test Outcome` == "Failure") %>%
  count(`Missile Name`)

# count total tests
ncount_total = mis.df %>%
  count(`Missile Name`)

# inner join onto the failed data
ncounts = ncount_fails %>%
  inner_join(ncount_total, by = "Missile Name")

# Grab the missile types to join with this data
types = mis.df %>%
  select(`Missile Name`, `Missile Type`) %>%
  distinct()

# Join missile types onto the data
ncounts = ncounts %>%
  left_join(types, by = "Missile Name")

# rename new count columns
```

```
ncounts = rename(ncounts, Failed = n.x, Tests = n.y)
# calculate failure rate
ncounts = ncounts %>%
  mutate(`Fail Rate` = round((Failed / Tests ) * 100, 2))

kable(ncounts %>%
        select(`Missile Name`, `Missile Type`, Failed, Tests, `Fail Rate`) %>%
        arrange(desc(`Fail Rate`)))
```

Missile Name	Missile Type	Failed	Tests	Fail Rate
Taepodong-1	SLV	1	1	100.00
Unha	SLV	2	2	100.00
Musudan	IRBM	7	8	87.50
Pukguksong-1	SLBM	3	6	50.00
Hwasong-12	IRBM	3	7	42.86
Scud-B MaRV	SRBM	1	3	33.33
Unha-3	SLV	1	3	33.33
Scud-B	SRBM	3	10	30.00
Nodong	MRBM	4	16	25.00
Unknown	Unknown	5	25	20.00
Unknown	SRBM	5	25	20.00
Unknown	ICBM	5	25	20.00
Unknown	SLBM	5	25	20.00
ER Scud	MRBM	1	8	12.50
KN-25	SRBM	1	23	4.35

Note the above only lists missiles where failures have been recorded. This lower level of detail paints an interesting picture. We can precisely identify, at least as far as the data is accurate, what specific missiles seem to be highly reliable, and which ones are clearly still early on in the development stages.

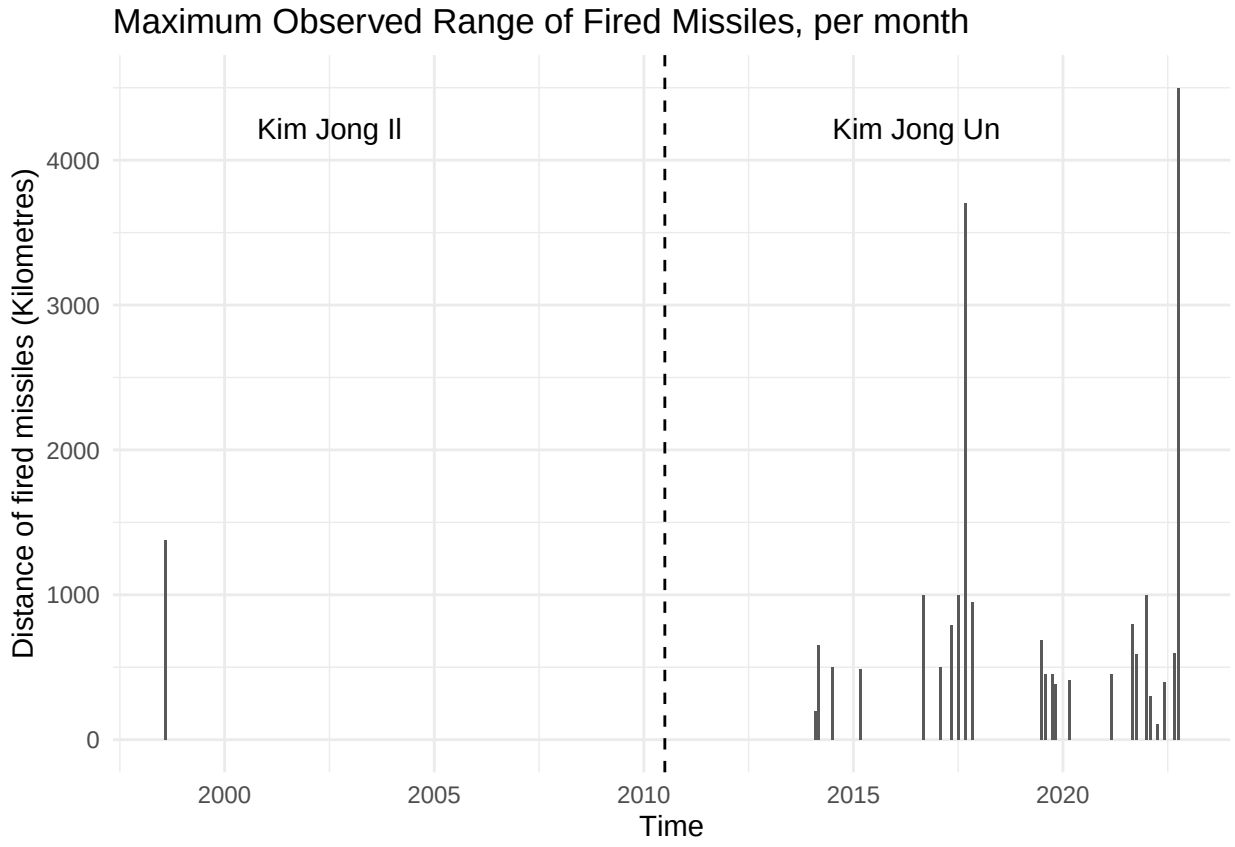
We see that as of the time of this data, no Taepodong-1 or Unha (both SLVs) had been successfully fired. However note that while the Taepodong-1 was technically a failure, it flew over 1000km, which we'll discuss soon. And we see they are struggling with the Musudan as well. We can see that the Pukguksong-1 is the least reliable SLBM where about half are failing, however we don't have data on other SLBMs. From there, we see the fail rate drop down to around 20-30% then down to the ER Scud and KN-25 which has about a 4% failure rate.

So while previously we had identified that the SLV capability required some work, we can see that the Unha-3 appears to be their better performing SLV, noting the small sample we are measuring here.

Missile test distances achieved

We've seen a significant increase in the number of missiles tested as well as much varied use of facilities in more recent years. While technical knowledge of missile systems is important, let's now look at the ranges successful missiles have travelled, and see how this may or may not have varied over time. This should give us a reasonable appreciation of any development of a long range strike capability. This time, we're grouping the data by month instead of by year so that we can get a bit more granularity and also we are aggregating by grabbing the maximum range for each month. In previous plots we were simply counting the number of events.

```
mis.df %>%
  group_by(month=floor_date(date_launch, "month")) %>%
  summarise(max_range = max(distance)) %>%
  ggplot(aes(month, max_range)) +
      # the following years are one year earlier than actual date due to way plot renders
  #geom_vline(aes(xintercept = as.Date("1993", format = "%Y")), lty = 2)+
  #geom_rect(aes(xmin=year[1], xmax=year[6], ymin=0, ymax=Inf, fill='yellow'))+
  geom_vline(aes(xintercept = as.Date("2010", format = "%Y")), lty = 2)+
  #geom_text("hello", )
  #annotate("text", x = as.Date("1991", format = "%Y"), y = 4000, label =  "Kim Il Sung")+
  annotate("text", x = as.Date("2002", format = "%Y"), y = 4220, label =  "Kim Jong Il")+
  annotate("text", x = as.Date("2016", format = "%Y"), y = 4220, label =  "Kim Jong Un")+
  geom_bar(stat = 'identity')+
  labs(title = "Maximum Observed Range of Fired Missiles, per month",
       x = "Time",
       y = "Distance of fired missiles (Kilometres)")+
  theme_minimal()
```



We can see an earlier significant test by Kim Jong Il, however the increase in tempo and range under Kim Jong Un is again clearly apparent. Under Un, the tested range of North Korean missiles has increased. Let's look at the tests where the missile flew over 1000km. We're sorting by the distance and including a number of relevant variables, like *Test Outcome*.

```
kable(mis.df %>%
  filter(distance > 1000) %>%
```

```
  select(date_launch, `Missile Name`, `Missile Type`, `Facility Name`, distance, `Test Outcome`) %>%
  arrange(desc(distance)))
```

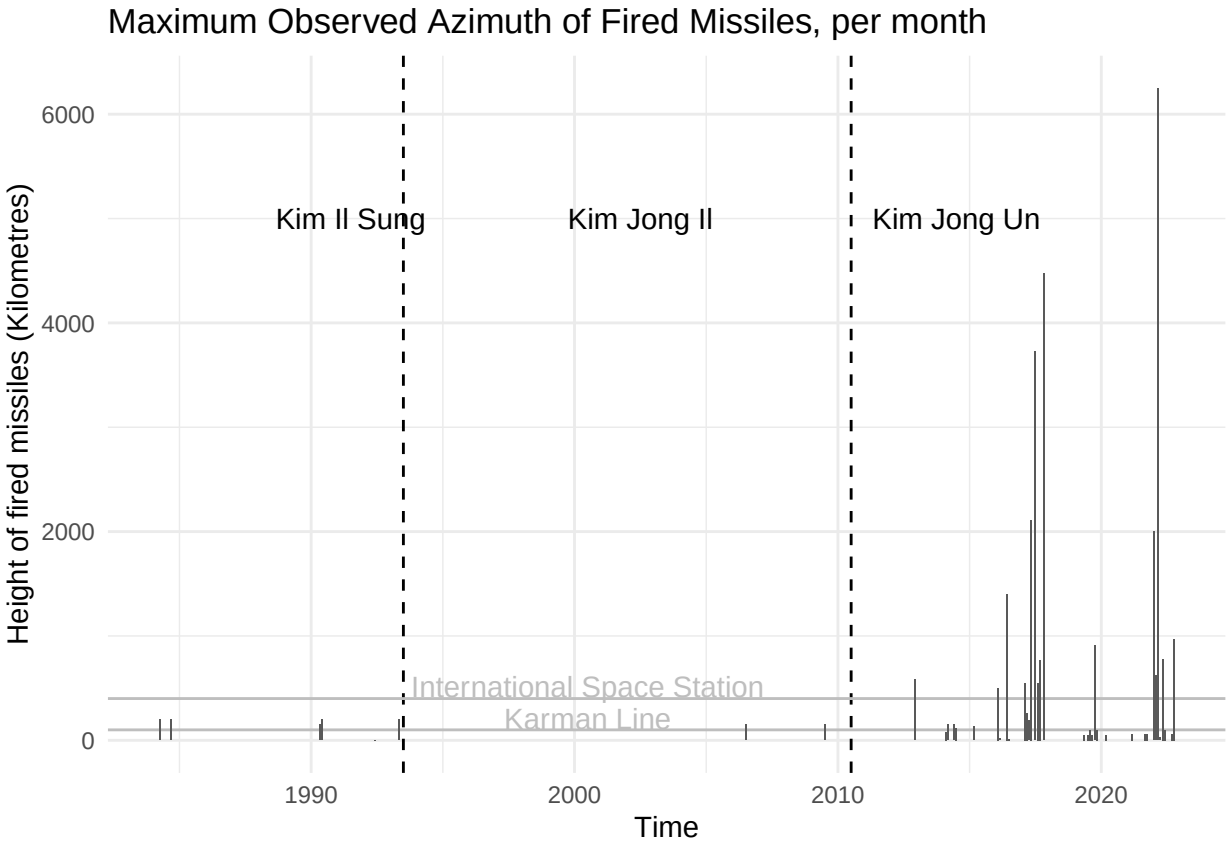
date_launch	Missile Name	Missile Type	Facility Name	distance	Test Outcome
2022-10-03	New IRBM (2022)	IRBM	Mupyong-ni Arms Plant	4500	Success
2017-09-14	Hwasong-12	IRBM	Pyongyang International Airport	3700	Success
2017-08-28	Hwasong-12	IRBM	Pyongyang International Airport	2700	Unknown
1998-08-31	Taepodong-1	SLV	Tonghae Satellite Launching Ground	1380	Failure
2022-03-24	Hwasong-17	ICBM	Pyongyang International Airport	1090	Success

A point worth noting here is that there are two events over 1000m in 2017 within one month of each other. However in the previous plot, we can only see the one entry, which was the larger and later one for 3700m. So this illustrates the importance of understanding how you may be aggregating data, as within an aggregation (especially one like maximum or minimum) you may miss out on important events that were close or similar to the one event chosen for a given time period.

Missile test heights achieved

We'll now conduct a similar analysis though looking at the height / azumith / elevation achieved.

```
mis.df %>%
  filter(!is.na(height)) %>% # Needed! Otherwise misses important events if they are close to NA values
  select(date_launch, height) %>%
  group_by(month=floor_date(date_launch, "month")) %>%
  summarise(max_height = max(height)) %>%
  ggplot(aes(month, max_height)) +
  #xlim(as.Date("1989", format = "%Y"),as.Date("2024", format = "%Y"))+
  geom_vline(aes(xintercept = as.Date("1993", format = "%Y")), lty = 2)+
  #geom_rect(aes(xmin=year[1], xmax=year[6], ymin=0, ymax=Inf, fill='yellow'))+
  geom_vline(aes(xintercept = as.Date("2010", format = "%Y")), lty = 2)+
  geom_hline(aes(yintercept = 100), color = 'grey')+
  annotate("text", x = as.Date("2000", format = "%Y"), y = 210, label =  "Karman Line", color = 'grey')+
  geom_hline(aes(yintercept = 400), color = 'grey')+
  annotate("text", x = as.Date("2000", format = "%Y"), y = 520, label =  "International Space Station",
  annotate("text", x = as.Date("1991", format = "%Y"), y = 5000, label =  "Kim Il Sung")+
  annotate("text", x = as.Date("2002", format = "%Y"), y = 5000, label =  "Kim Jong Il")+
  annotate("text", x = as.Date("2014", format = "%Y"), y = 5000, label =  "Kim Jong Un")+
  geom_bar(stat = 'identity')+
  labs(title = "Maximum Observed Azimuth of Fired Missiles, per month",
       x = "Time",
       y = "Height of fired missiles (Kilometres)")+
  theme_minimal()
```



All of the missile tests with high azimuth / elevation were conducted during the time of Kim Jong Un. We see some very high firings indeed. Note the horizontal reference lines. The *Karman Line* is the main generally accepted boundary between the earth's atmosphere and space. The International Space Station orbits the planet somewhere between 300-400km above the earth. We can see that under Un, the elevation of missiles tested in North Korea has increased. These missile tests were typically fired very high, but not comparatively far in distance. However it is expected that due to the significant elevations achieved, North Korea could fire significantly further distances using flatter trajectories than what has been tested.

Let's now look at the missiles that were used where the elevation achieved was above 500km (so higher than the international space station), sorted by height.

```
kable(mis.df %>%
  filter(height > 500) %>%
  select(date_launch, `Missile Name`, `Missile Type`, `Facility Name`, height, `Test Outcome`) %>%
  arrange(desc(height)))
```

date_launch	Missile Name	Missile Type	Facility Name	height	Test Outcome
2022-03-24	Hwasong-17	ICBM	Pyongyang International Airport	6248	Success
2017-11-28	Hwasong-15	ICBM	Pyongsong Field	4475	Success
2017-07-28	Hwasong-14	ICBM	Mupyong-ni Arms Plant	3724	Success
2017-07-04	Hwasong-14	ICBM	Panghyon	2802	Success
2017-05-14	Hwasong-12	IRBM	North Kusong Testing Ground	2111	Success
2022-01-29	Hwasong-12	IRBM	Mupyong-ni Arms Plant	2000	Success

date_launch	Missile Name	Missile Type	Facility Name	height	Test Outcome
2016-06-21	Musudan	IRBM	Wonsan Kalma International Airport	1400	Success
2022-10-03	New IRBM (2022)	IRBM	Mupyong-ni Arms Plant	970	Success
2019-10-01	Pukguksong-3	SLBM	Yonghung Bay	910	Success
2022-05-04	Unknown	ICBM	Pyongyang International Airport	780	Failure
2017-09-14	Hwasong-12	IRBM	Pyongyang International Airport	770	Success
2022-02-26	Unknown	Unknown	Pyongyang International Airport	620	Success
2012-12-12	Unha-3	SLV	Sohae Satellite Launching Station	581	Success
2017-05-21	Pukguksong-2	MRBM	Lake Yonpung	560	Success
2022-03-04	Unknown	Unknown	Pyongyang International Airport	560	Success
2017-02-11	Pukguksong-2	MRBM	Kusong Testing Ground	550	Success
2017-08-28	Hwasong-12	IRBM	Pyongyang International Airport	550	Unknown
2016-02-07	Unha-3	SLV	Sohae Satellite Launching Station	502	Success

We can see that most of those missiles were IRBM or ICBM missiles, and what is noteworthy is that they were almost all successful. We see some SLVs, a SLBM and MRBMs as well. The ICBM and IRBMs achieved the greatest elevation.

There aren't too many patterns of note regarding the launch facilities used. We again see the Pyongyang International Airport feature heavily, including the recent test of an ICBM which achieved over 6,000km in elevation. While it hasn't come to our attention so far, the Mupyong-ni Arms Plant seems like it may be significant seeing as it was the facility used to launch an ICMB to over 3,000km in 2017 and an IRBM to over 2,000km in 2022.

Not all noteworthy missile tests had heights or elevations recorded against them, mainly as they happened some time ago. However they do tend to have the impact area they are suspected to have landed. We'll look at the missile tests where the landing location was recorded as being in the Pacific Ocean. This is significant as this indicates the missiles flew over Japan. Note in the filter we've noticed that there are two landing locations indicating landings in the Pacific Ocean. This comes back to really exploring your data as you're going through this process.

```
kable(mis.df %>%
  filter(`Landing Location` == "Pacific Ocean" | `Landing Location` == "330km east of Hachinohe and 400
  select(date_launch, `Missile Name`, `Missile Type`, `Facility Name`, `Test Outcome`))
```

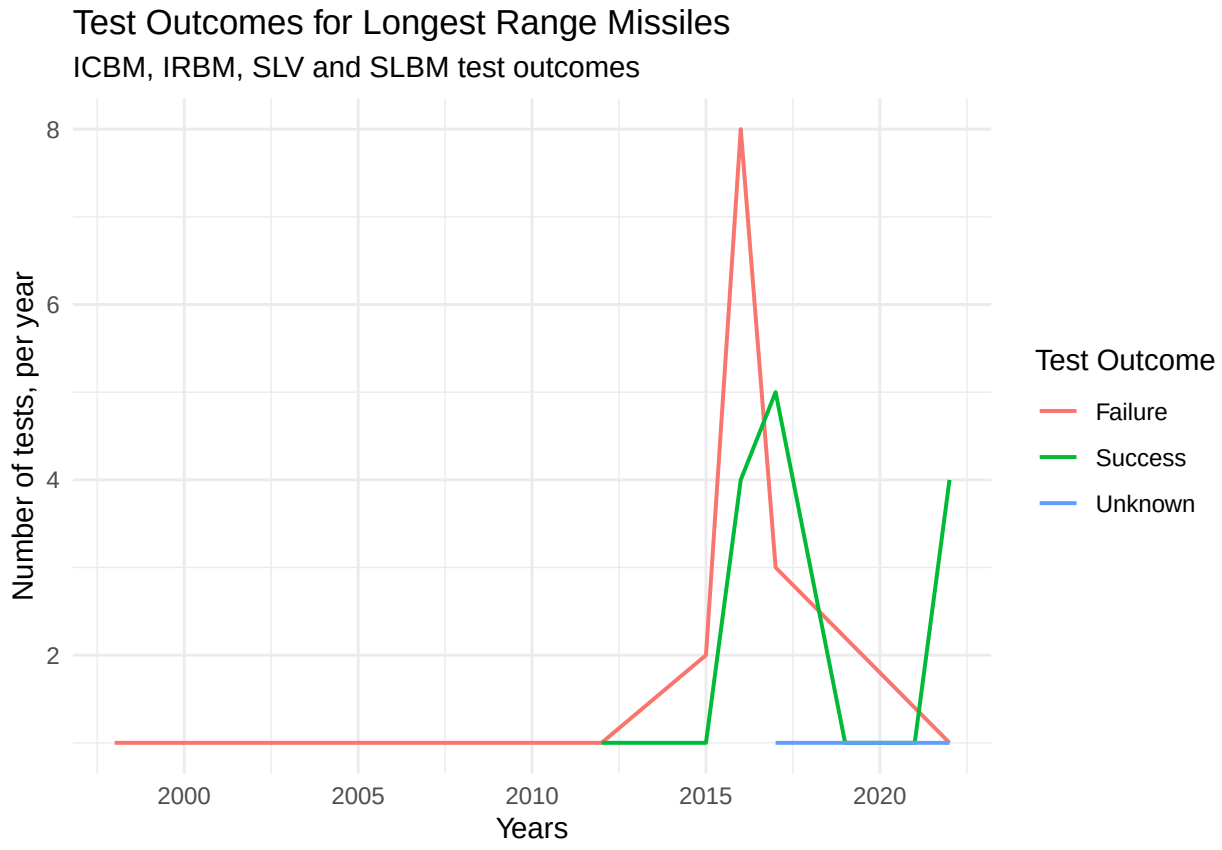
date_launch	Missile Name	Missile Type	Facility Name	Test Outcome
1998-08-31	Taepodong-1	SLV	Tonghae Satellite Launching Ground	Failure
2009-04-05	Unha	SLV	Tonghae Satellite Launching Ground	Failure
2012-04-13	Unha-3	SLV	Sohae Satellite Launching Station	Failure
2017-08-28	Hwasong-12	IRBM	Pyongyang International Airport	Unknown
2017-09-14	Hwasong-12	IRBM	Pyongyang International Airport	Success
2022-10-03	New IRBM (2022)	IRBM	Mupyong-ni Arms Plant	Success

While the overall number of missiles fired in the Pacific Ocean is low, they have increased in capability from three initial failures to the two most recent tests being successes. This may also be down to the IRBM platform being more reliable than the SLV platform.

Are failure rates for the longest range missiles changing?

Are the failure rates of North Koreas longest range strike missiles improving (lowering) over time? This would provide some indication as to whether the threat from North Korean missiles is changing.

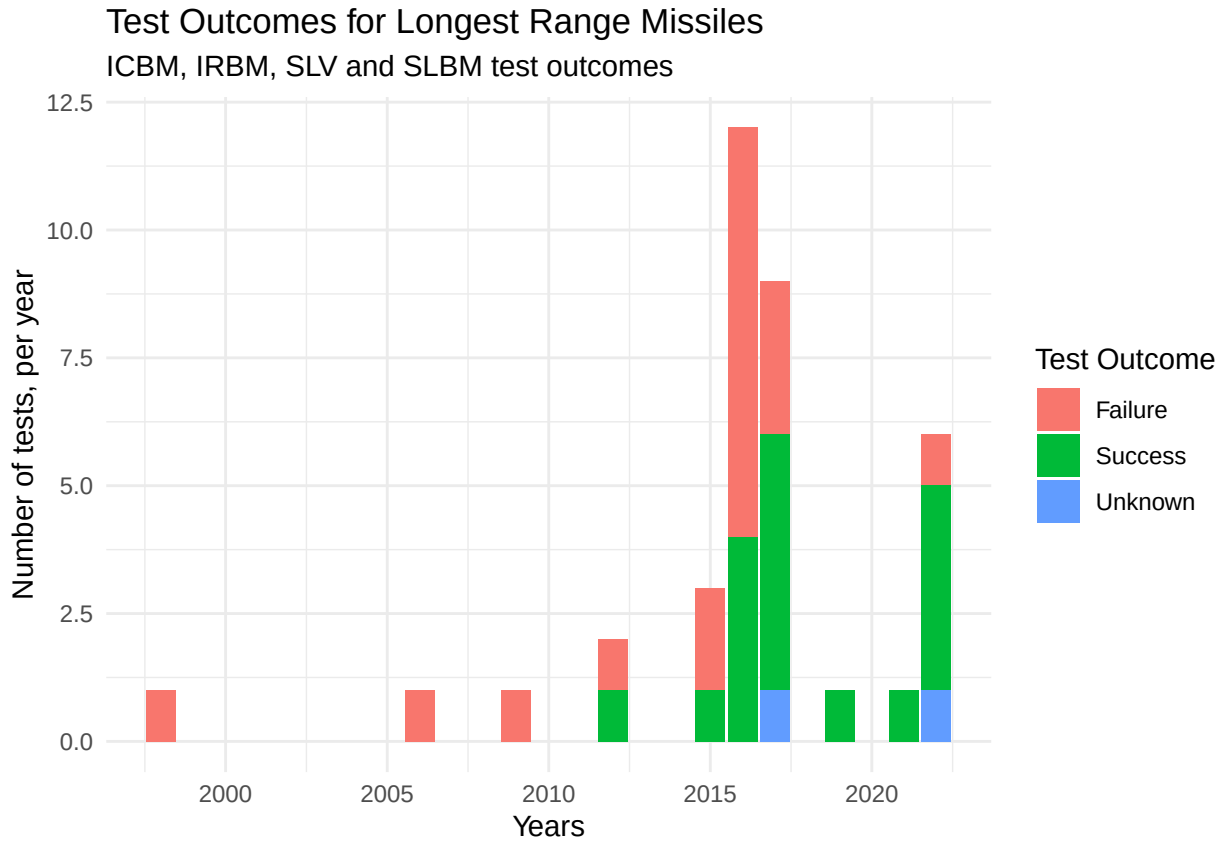
```
mis.df %>%
  select(date_launch, `Missile Type`, `Test Outcome`) %>%
  filter(`Missile Type` == "SLV" | `Missile Type` == "IRBM" | `Missile Type` == "ICBM" | `Missile Type`
  group_by(year=floor_date(date_launch, "year"), `Test Outcome`) %>%
  summarise(sumz = n()) %>%
  # sum(`Test Outcome` == "Failure")
  #select(year, `Test Outcome`) %>%
  ggplot(aes(year, sumz, col = `Test Outcome`)) +
  geom_line(linewidth = 0.7) +
  labs(title = "Test Outcomes for Longest Range Missiles",
       subtitle = "ICBM, IRBM, SLV and SLBM test outcomes",
       x = "Years",
       y = "Number of tests, per year")+
  theme_minimal()
```



We'll also try to use a bar plot to see if there is a clearer visualisation.

```
mis.df %>%
  select(date_launch, `Missile Type`, `Test Outcome`) %>%
  filter(`Missile Type` == "SLV" | `Missile Type` == "IRBM" | `Missile Type` == "ICBM" | `Missile Type`
  group_by(year=floor_date(date_launch, "year"), `Test Outcome`) %>%
```

```
summarise(sumz = n()) %>%
# sum(`Test Outcome` == "Failure")
#select(year, `Test Outcome`) %>%
ggplot(aes(year, sumz, fill = `Test Outcome`)) +
geom_bar(stat = 'identity') +
labs(title = "Test Outcomes for Longest Range Missiles",
     subtitle = "ICBM, IRBM, SLV and SLBM test outcomes",
     x = "Years",
     y = "Number of tests, per year")+
theme_minimal()
```



Due to the zero values and the small number of measures being compared, just three, the bar plot here provides clearer insights.

The above plot seems to indicate that after a large number of tests around the 2016 period, which included many failures, the number of failures has decreased and the number of successes has increased - especially as a proportion of the whole. This indicates that North Korea is becoming more capable in testing it's longest range missiles.

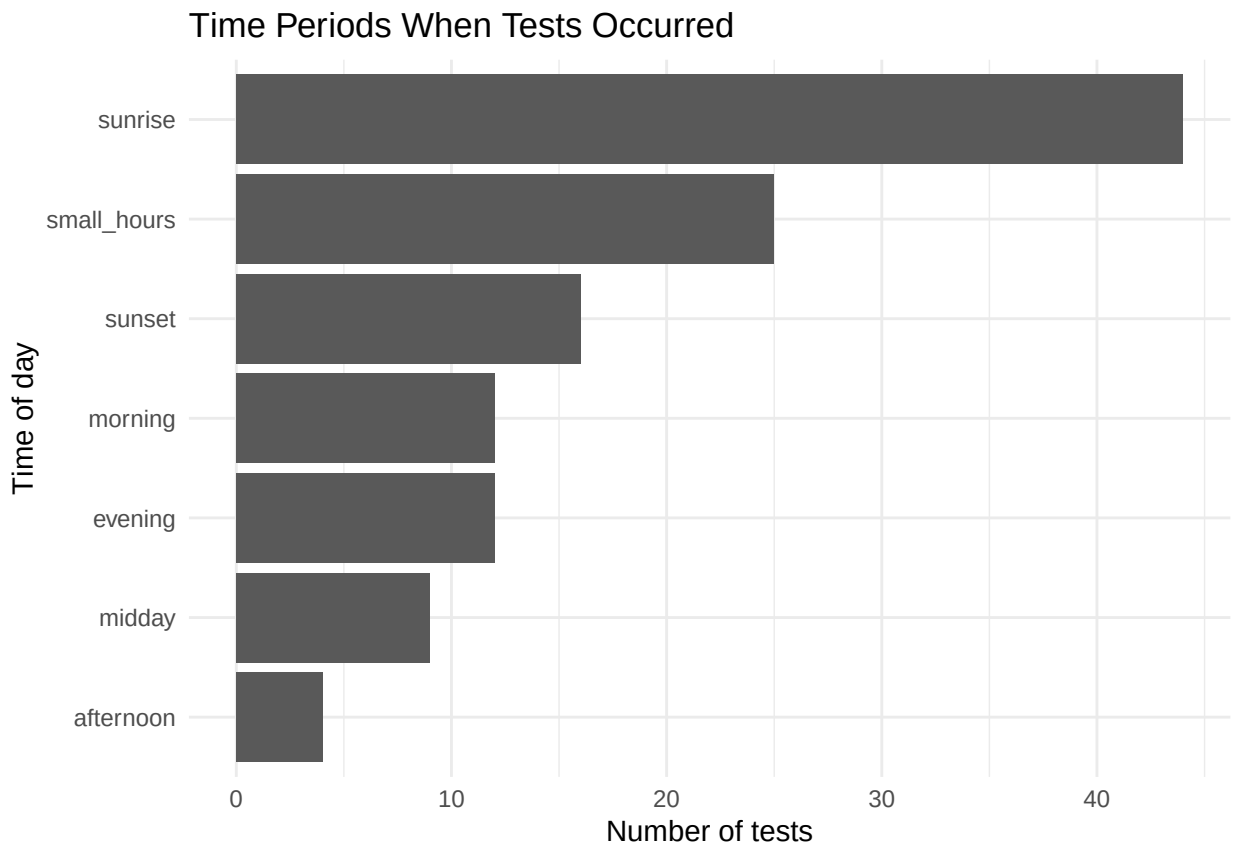
Daily patterns?

Let's see if there seems to be any preference for first half of the day or last half, to see if we should look deeper into time of day preferences.

We must also note some limitations in our data that a lot of historic tests do not contain any time data so this sort of analysis will only be relevant for more recent tests, which is fine in this case. Our original data

was in UTC timezone so we previously convert these times to be KST, by adding 9hrs.

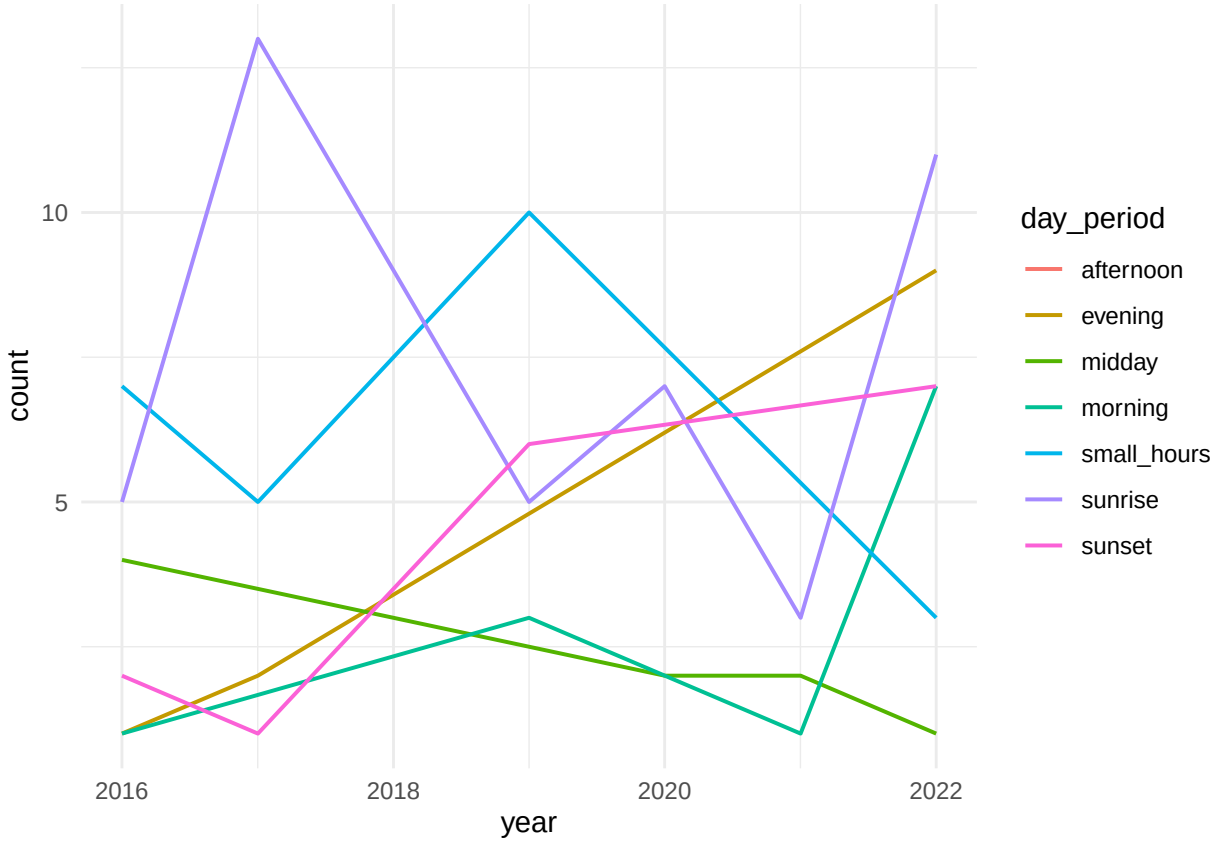
```
# need to make sure original UTC time is converted to local KST timezone first
mis.df = mis.df %>%
  mutate(day_period = case_when(
    hour(dateandtime) < 6 ~ "small_hours",
    hour(dateandtime) < 8 ~ "sunrise",
    hour(dateandtime) < 11 ~ "morning",
    hour(dateandtime) < 13 ~ "midday",
    hour(dateandtime) < 16 ~ "afternoon",
    hour(dateandtime) < 20 ~ "sunset",
    hour(dateandtime) < 24 ~ "evening",
    TRUE ~ "nil"
  ))

mis.df %>%
  filter(day_period != "nil") %>%
  count(day_period) %>%
  #arrange(desc(n))
  ggplot(aes(reorder(day_period, n), n))+
  coord_flip()+
  geom_bar(stat = 'identity')+
  labs(title = "Time Periods When Tests Occurred",
       y = "Number of tests",
       x = "Time of day")+
  theme_minimal()
```



From the above plot, we can see a heavy preference for conducting tests in the early morning, either at sunrise or before it in the small hours.

We'll quickly look to see if there have been any changes over time.

```
mis.df %>%
  group_by(year=floor_date(date_launch, "year"), day_period) %>%
  filter(day_period != "nil") %>%
  summarise(count = n()) %>%
  ggplot(aes(year, count, col = day_period)) +
  geom_line(linewidth = 0.7)+
  theme_minimal()
```



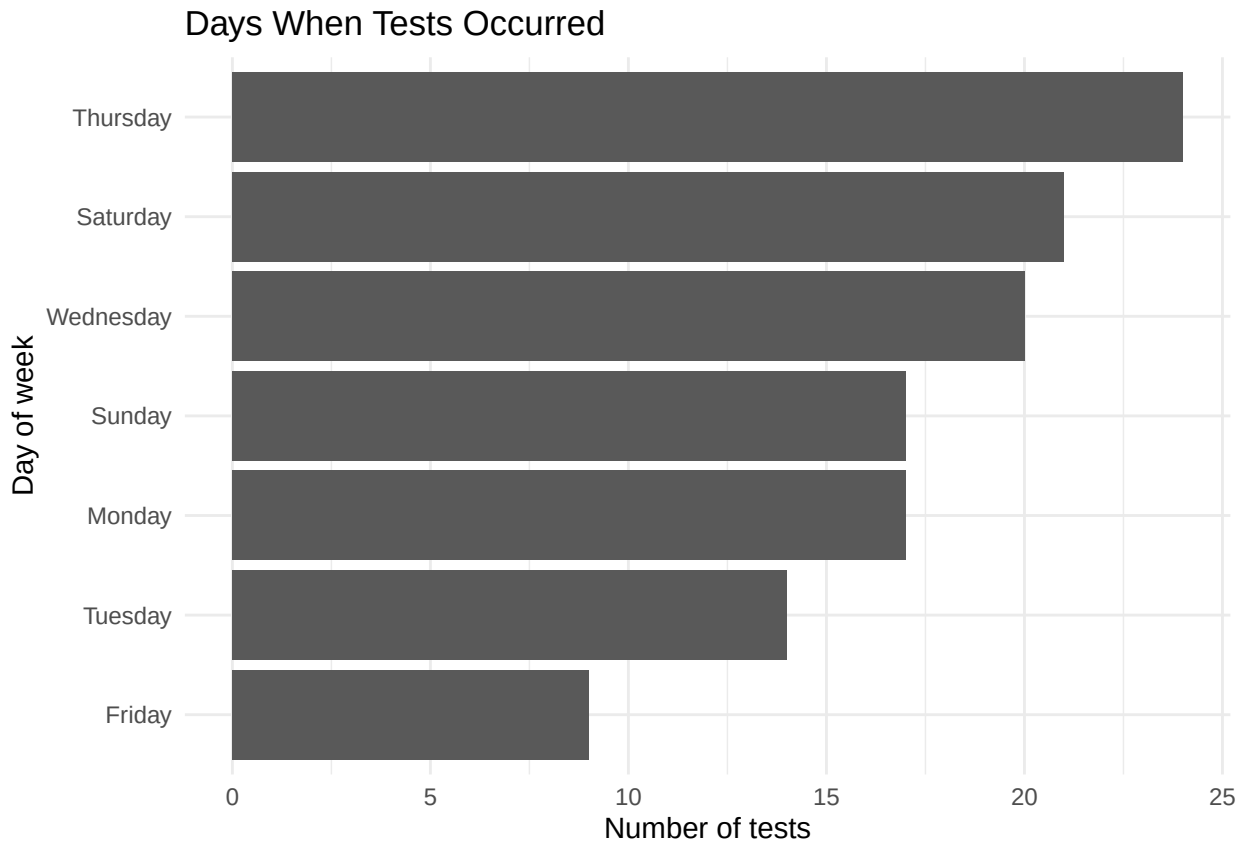
The above plot is rather messy. For the most part, the day periods have been mostly fairly consistent. However we have seen a general upward trend for tests conducted in the evening and sunset, and a decrease in small hours. Perhaps Kim Jong Un isn't liking early starts? Or perhaps there are a range of other factors, like weather. Either way, this could prompt additional qualitative research to explore further.

Days of the week

We'll do the same thing, but for the days of the week.

```
mis.df$weekday <- weekdays(mis.df$dateandtime)

mis.df %>%
```

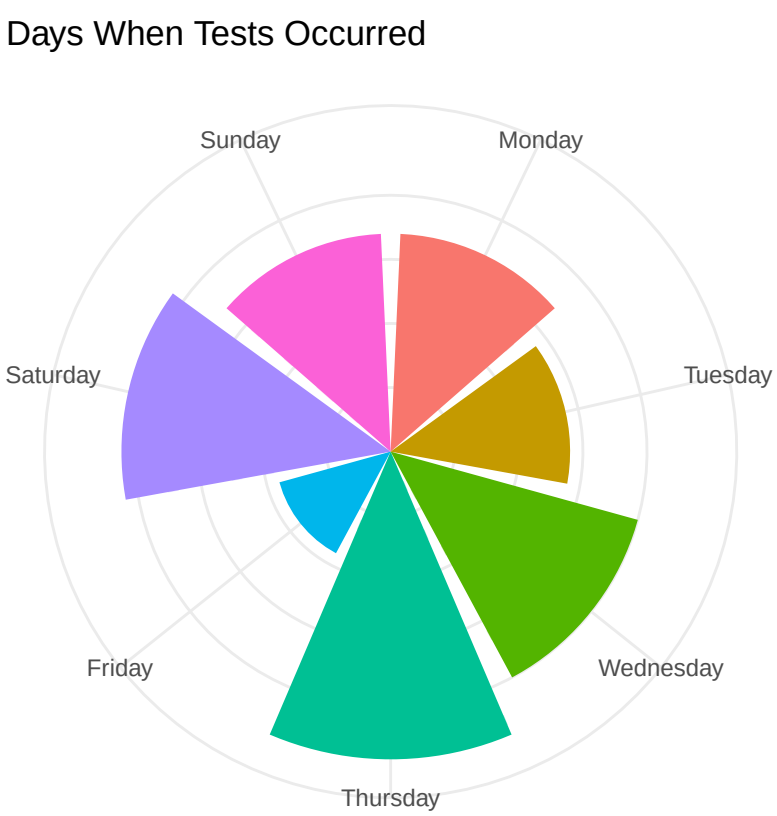
```
  count(weekday) %>%
  filter(!is.na(weekday)) %>%
  ggplot(aes(reorder(weekday, n), n))+
  coord_flip()+
  geom_bar(stat = 'identity')+
  labs(title = "Days When Tests Occurred",
       y = "Number of tests",
       x = "Day of week")+
  theme_minimal()
```



We'll also see if a polar plot provides a better visualisation. While the above plot sorted the days by frequency, in the below we'll keep the days in week order. Note it can be considered bad practise to sort days of the week as days should always be left in week order as there is an inherent order to days. However, it can be useful to look for patterns by day. Sorting them can help. Alternatively, you could use the above bar plot but just keep the days in normal week order.

```
# experimenting with a polar plot
mis.df %>%
  count(weekday) %>%
  filter(!is.na(weekday)) %>%
  mutate(weekday = factor(weekday, levels = c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "S
  ggplot() +
  geom_col(aes(x = weekday, fill = weekday, y = n))+
  coord_polar()+
  #scale_fill_gradient(low='red', high='white', limits = c(2,30))+
  #scale_fill_discrete(breaks = c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "
```

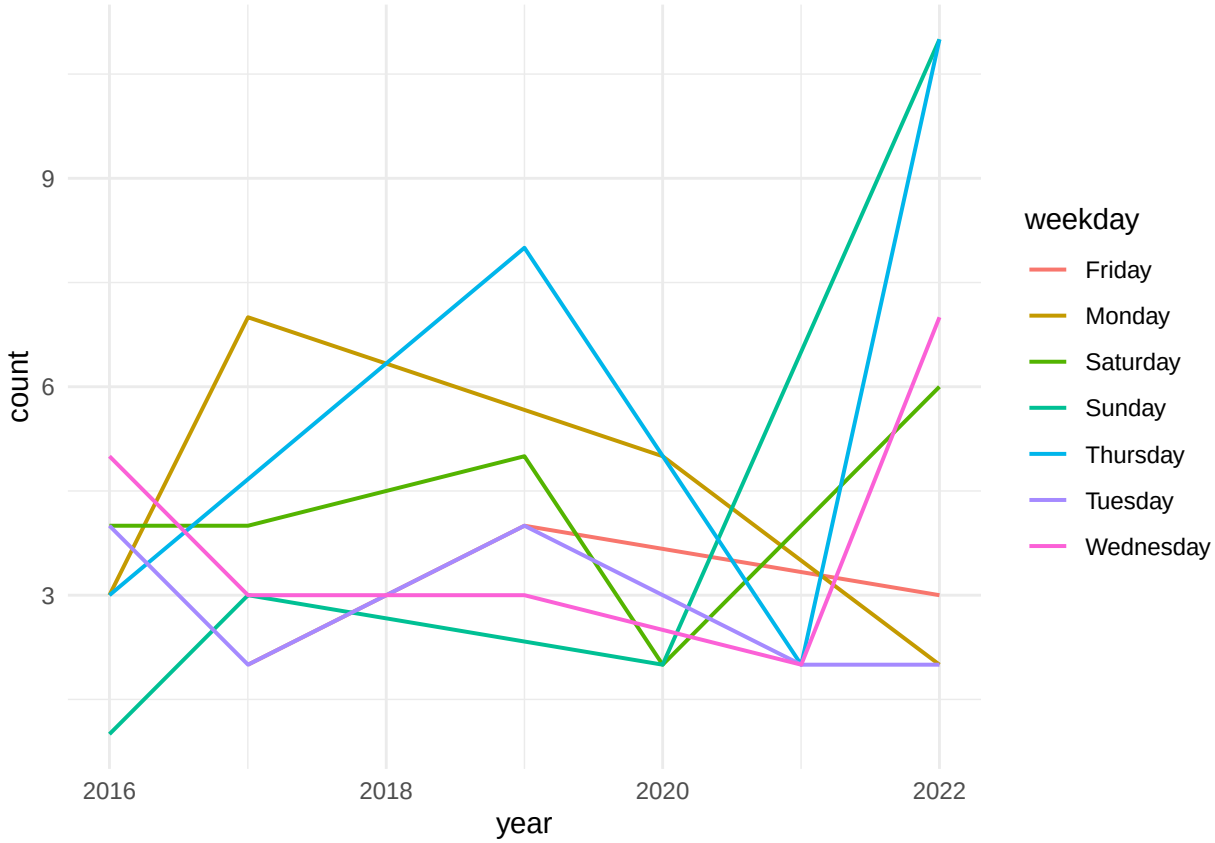
```
  labs(title = "Days When Tests Occurred",
       x = NULL,
       y = NULL)+
  theme_minimal()+
  theme(legend.position = "none", axis.text.y = element_blank())
```



There doesn't seem to be any strong preference of the time of the day in which to conduct tests. While there don't seem to be any strong standouts, we can see at least that friday sees far fewer launches than Thursday. This could simply be a coincidence, or it may not. This could justify further qualitative research to explore this. The launch dates are likely more heavily influenced by the weather than any weekly routine, but it can still be good to know.

We'll quickly look to see if there have been any changes over time.

```
mis.df %>%
  group_by(year=floor_date(date_launch, "year"), weekday) %>%
  filter(!is.na(weekday)) %>%
  summarise(count = n()) %>%
  ggplot(aes(year, count, col = weekday)) +
  geom_line(linewidth = 0.7)+
  theme_minimal()
```



From the above plot, we can see that for the most part, the weekdays have been fairly consistent with some seemingly random variance. We do see that Sunday and Thursday saw large increases in 2022. This plot isn't too informative by itself, but we do have the data now to look for any potentially significant changes.

Multiple launches on one day?

How often does North Korea conduct multiple missile tests on one day? What is the maximum number of tests North Korea has conducted in one day?

```
kable(mis.df %>%
  count(date_launch) %>%
  arrange(desc(n)) %>%
  head(10))
```

date_launch	n
2022-06-05	8
2006-07-05	7
2009-07-05	7
2015-02-08	5
2017-03-05	5
2014-02-27	4
2015-04-03	4
1984-04-09	3
1984-09-01	3

date_launch	n
1993-05-29	3

Let's look at the launches that happened on 5 Jun 2022, as it's the most recent and the most launches on one day.

```
kable(mis.df %>%
  filter(date_launch == "2022-06-05") %>%
  select(dateandtime, `Missile Name`, `Missile Type`, `Facility Name`, `Landing Location`, height, dista
```

dateandtime	Missile Name	Missile Type	Facility Name	Landing Location	height	distance
2022-06-05 21:06:00	Unknown	SRBM	Unknown	Sea of Japan or East Sea	50	350
2022-06-05 21:10:00	Unknown	SRBM	Unknown	Sea of Japan or East Sea	50	300
2022-06-05 21:15:00	Unknown	SRBM	Unknown	Sea of Japan or East Sea	50	400
NA	Unknown	SRBM	Unknown	Sea of Japan or East Sea	57	390
2022-06-05 21:24:00	Unknown	SRBM	Unknown	Sea of Japan or East Sea	100	350
2022-06-05 21:30:00	Unknown	SRBM	Unknown	Sea of Japan or East Sea	50	400
NA	Unknown	SRBM	Unknown	Sea of Japan or East Sea	57	390
2022-06-05 21:41:00	Unknown	SRBM	Unknown	Sea of Japan or East Sea	100	300

In this dataset, the launch location (Facility Name) is not known. Exploring the data one can see that many variables in these launches are unknown. However, it appears as though these were all SRBMs and all were fired into the Sea of Japan. Of note, reading the field that contain additional information, news sources indicate that the firings were conducted from multiple locations. It simply appears that the precise locations for each firing may not be known. This explains why the Facility Name fields were *Unknown*. Also note the firings were all conducted in sequence in roughly five minute intervals.

Here are the descriptions of the above launches.

```
mis.df %>%
  filter(date_launch == "2022-06-05") %>%
  select(`Additional Information`)
```

```
##
## 1 On June 5, North Korea launched several short-range ballistic missiles from multiple locations in
## 2
## 3
## 4
## 5
## 6
## 7
## 8
```

We'll also look at the launches that occurred on 2017, noting this was the time prior to 2022 that multiple launches happened on one day.

```
kable(mis.df %>%
  filter(date_launch == "2017-03-05") %>%
  select(dateandtime, `Missile Name`, `Missile Type`, `Facility Name`, `Landing Location`, height, dista
```

dateandtime	Missile Name	Missile Type	Facility Name	Landing Location	height	distance
2017-03-06 07:34:00	ER Scud	MRBM	Sohae Satellite Launching Station	Sea of Japan or East Sea	260	1000
2017-03-06 07:34:00	ER Scud	MRBM	Sohae Satellite Launching Station	Sea of Japan or East Sea	260	1000
2017-03-06 07:34:00	ER Scud	MRBM	Sohae Satellite Launching Station	Sea of Japan or East Sea	260	1000
2017-03-06 07:34:00	ER Scud	MRBM	Sohae Satellite Launching Station	Sea of Japan or East Sea	260	1000
2017-03-06 07:34:00	ER Scud	MRBM	Sohae Satellite Launching Station		NA	NA

Interestingly, these were all ER Scud MRBMs, all again launched in quick succession - this time seemingly at the same time. We can see that on both of these occasions, only one missile was being tested. In the 2017 instance, they were all fired from the one facility. In the more recent 2022 case, multiple sites were used. This could provide another indication of gradually increasing sophistication, coordination and complexity in North Korea's missile testing and capability.

We'll also look at the two 2015 cases.

```
kable(mis.df %>%
  filter(date_launch == "2015-04-03" | date_launch == "2015-02-08") %>%
  select(date_launch, `Missile Name`, `Missile Type`, `Facility Name`, `Landing Location`, height, dista
```

date_launch	Missile Name	Missile Type	Facility Name	Landing Location	height	distance
2015-02-08	KN-02	SRBM	Hodo Peninsula	Sea of Japan or East Sea	NA	NA
2015-02-08	KN-02	SRBM	Hodo Peninsula	Sea of Japan or East Sea	NA	NA
2015-02-08	KN-02	SRBM	Hodo Peninsula	Sea of Japan or East Sea	NA	NA
2015-02-08	KN-02	SRBM	Hodo Peninsula	Sea of Japan or East Sea	NA	NA
2015-02-08	KN-02	SRBM	Hodo Peninsula	Sea of Japan or East Sea	NA	NA
2015-04-03	KN-02	SRBM	Hodo Peninsula	Sea of Japan or East Sea	NA	NA
2015-04-03	KN-02	SRBM	Hodo Peninsula	Sea of Japan or East Sea	NA	NA
2015-04-03	KN-02	SRBM	Hodo Peninsula	Sea of Japan or East Sea	NA	NA

date_launch	Missile Name	Missile Type	Facility Name	Landing Location	height	distance
2015-04-03	KN-02	SRBM	Hodo Peninsula	Sea of Japan or East Sea	NA	NA

We don't actually have a lot of data about these launches. However they do again seem to have been multiple firings of all the same missiles, this time being KN-02 SRBMs. These were all fired from the Hodo Peninsula facility.

This is an interesting historical pattern, same missiles on the same day. The one newer development has been the use of multiple locations.

Modelling

While there is more graphical data analysis that could be explored further, we'll now consider potential opportunities or interesting variables we may like to be able to predict.

Predicting distance

Could we rapidly predict the estimated *distance* a missile will travel by the facility and height? Could this potentially be used in real time?

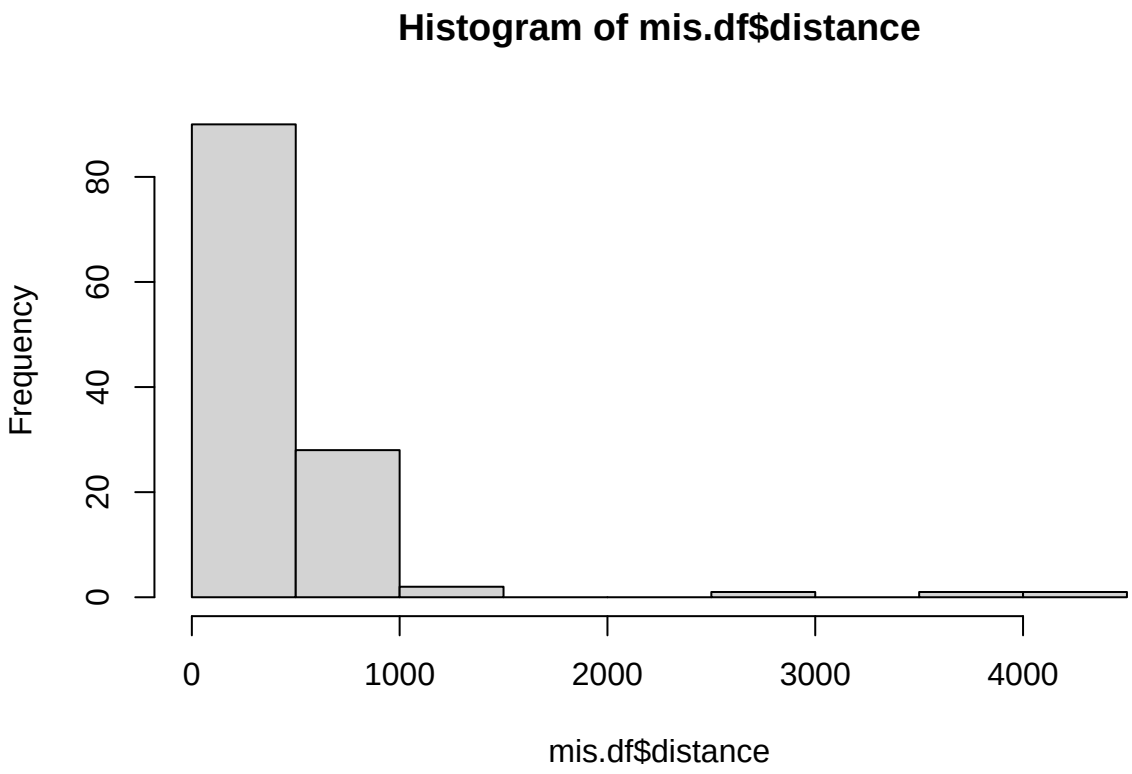
If we could detect the launch location as well as the height, could we predict the range? Would this be useful? Potentially. However we have already seen that some of the missile tests are firing at very high heights and this is likely due to testing. It would be considered that the trajectories for real missile attacks would be flatter. So would this be useful for real? Perhaps the SRBM and MRBM tests use flatter trajectories so would be more like real missile attacks? We should probably include the missile type in the model to account for this.

Here is what our proposed model looks like.

Distance ~ facility + missile type + height

The distance variable appears to have a roughly Poisson distribution.

```
hist(mis.df$distance)
```



However the mean is not identical to the variance.

```
#mean(!is.na(mis.df$distance))
#var(!is.na(mis.df$distance))
```

The mean (0.6) is larger than the variance (0.24). This indicates that we should conduct some transformations or tweak the model. We won't dive right into modelling in detail as the steps we'd need to take from here will get technical quickly and that is not the point of this example.

Poisson regression model

For basic demonstration purposes, we'll build a poisson generalised linear model and check the diagnostics.

```
dist.glm = glm(distance ~ `Facility Name` + `Missile Type` + height, data = mis.df, family = 'poisson')
summary(dist.glm)
```

```
##
## Call:
## glm(formula = distance ~ `Facility Name` + `Missile Type` + height,
##     family = "poisson", data = mis.df)
##
## Coefficients:
##                                                 Estimate Std. Error z value
## (Intercept)                                    5.104e+00  8.863e-02  57.585
```

```
## `Facility Name`Hodo Peninsula                       -6.177e-02  5.490e-02  -1.125
## `Facility Name`Hungnam                              -5.886e-01  5.564e-02 -10.579
## `Facility Name`Hwangju                               1.764e-01  5.815e-02   3.033
## `Facility Name`Kaechon Air Base                     -2.412e-01  6.242e-02  -3.864
## `Facility Name`Kaesong                               1.772e-01  5.815e-02   3.047
## `Facility Name`Kittaeryong Missile Base             -7.407e-01  8.591e-02  -8.622
## `Facility Name`Kusong Testing Ground                -3.260e-01  7.196e-02  -4.530
## `Facility Name`Kwail Airbase                         6.847e-02  5.909e-02   1.159
## `Facility Name`Lake Yonpung                         -3.256e-01  7.197e-02  -4.524
## `Facility Name`Munchon                              -1.807e-01  6.172e-02  -2.928
## `Facility Name`Mupyong-ni Arms Plant                 1.968e-01  5.349e-02   3.679
## `Facility Name`Nampo                                 1.575e-01  5.833e-02   2.701
## `Facility Name`North Kusong Testing Ground          -1.018e+00  6.425e-02 -15.846
## `Facility Name`Panghyon                              2.698e-01  6.414e-02   4.207
## `Facility Name`Pyongsong Field                       3.554e-01  6.474e-02   5.490
## `Facility Name`Pyongyang International Airport       2.124e-01  5.131e-02   4.140
## `Facility Name`Samsok                                3.151e-01  5.701e-02   5.527
## `Facility Name`Sangum-ri                            -5.602e-01  6.682e-02  -8.385
## `Facility Name`Sinpo Shipyard                       -4.367e+00  1.394e-01 -31.320
## `Facility Name`Sohae Satellite Launching Station     3.554e-01  5.827e-02   6.099
## `Facility Name`Sondok                               -7.419e-01  6.362e-02 -11.662
## `Facility Name`Sondok Airbase                       -3.909e-02  5.663e-02  -0.690
## `Facility Name`Sunan                                 6.899e-02  5.587e-02   1.235
## `Facility Name`Sunchon                              -1.335e-01  5.533e-02  -2.413
## `Facility Name`Sunchon Airbase                      -9.649e-02  5.629e-02  -1.714
## `Facility Name`Taechon Reservoir                    -4.358e+00  1.481e-01 -29.420
## `Facility Name`Tongchan                             -6.030e-01  6.749e-02  -8.934
## `Facility Name`Tonghae Satellite Launching Ground   -3.422e-01  7.167e-02  -4.774
## `Facility Name`Uiju                                  2.296e-02  5.953e-02   0.386
## `Facility Name`Unknown                              -2.315e-01  5.153e-02  -4.493
## `Facility Name`West Sunan                           -1.133e-01  6.094e-02  -1.860
## `Facility Name`Wonsan Kalma International Airport   -8.622e-01  5.792e-02 -14.886
## `Facility Name`Yangdok                               6.448e-01  5.483e-02  11.760
## `Facility Name`Yonghung Bay                         -4.612e+00  1.499e-01 -30.766
## `Facility Name`Yonpo Airport                        -3.909e-02  5.663e-02  -0.690
## `Missile Type`ICBM                                   1.578e+00  8.051e-02  19.599
## `Missile Type`IRBM                                   2.668e+00  7.270e-02  36.694
## `Missile Type`MRBM                                   1.459e+00  7.401e-02  19.712
## `Missile Type`SLBM                                   5.654e+00  1.509e-01  37.455
## `Missile Type`SRBM                                   9.383e-01  7.402e-02  12.677
## `Missile Type`Unknown                                6.278e-01  7.733e-02   8.119
## height                                              -4.039e-05  8.255e-06  -4.893
##                                                     Pr(>|z|)
## (Intercept)                                          < 2e-16 ***
## `Facility Name`Hodo Peninsula                       0.260497
## `Facility Name`Hungnam                               < 2e-16 ***
## `Facility Name`Hwangju                              0.002420 **
## `Facility Name`Kaechon Air Base                     0.000112 ***
## `Facility Name`Kaesong                              0.002311 **
## `Facility Name`Kittaeryong Missile Base              < 2e-16 ***
## `Facility Name`Kusong Testing Ground                5.89e-06 ***
## `Facility Name`Kwail Airbase                        0.246608
## `Facility Name`Lake Yonpung                         6.06e-06 ***
## `Facility Name`Munchon                              0.003414 **
```

```
## ‘Facility Name‘Mupyong-ni Arms Plant              0.000234 ***
## ‘Facility Name‘Nampo                             0.006911 **
## ‘Facility Name‘North Kusong Testing Ground        < 2e-16 ***
## ‘Facility Name‘Panghyon                          2.59e-05 ***
## ‘Facility Name‘Pyongsong Field                   4.01e-08 ***
## ‘Facility Name‘Pyongyang International Airport    3.48e-05 ***
## ‘Facility Name‘Samsok                            3.25e-08 ***
## ‘Facility Name‘Sangum-ri                         < 2e-16 ***
## ‘Facility Name‘Sinpo Shipyard                    < 2e-16 ***
## ‘Facility Name‘Sohae Satellite Launching Station  1.07e-09 ***
## ‘Facility Name‘Sondok                            < 2e-16 ***
## ‘Facility Name‘Sondok Airbase                    0.490045
## ‘Facility Name‘Sunan                             0.216892
## ‘Facility Name‘Sunchon                           0.015803 *
## ‘Facility Name‘Sunchon Airbase                   0.086509 .
## ‘Facility Name‘Taechon Reservoir                 < 2e-16 ***
## ‘Facility Name‘Tongchan                          < 2e-16 ***
## ‘Facility Name‘Tonghae Satellite Launching Ground 1.80e-06 ***
## ‘Facility Name‘Uiju                              0.699663
## ‘Facility Name‘Unknown                           7.03e-06 ***
## ‘Facility Name‘West Sunan                        0.062953 .
## ‘Facility Name‘Wonsan Kalma International Airport  < 2e-16 ***
## ‘Facility Name‘Yangdok                           < 2e-16 ***
## ‘Facility Name‘Yonghung Bay                      < 2e-16 ***
## ‘Facility Name‘Yonpo Airport                     0.490045
## ‘Missile Type‘ICBM                               < 2e-16 ***
## ‘Missile Type‘IRBM                               < 2e-16 ***
## ‘Missile Type‘MRBM                               < 2e-16 ***
## ‘Missile Type‘SLBM                               < 2e-16 ***
## ‘Missile Type‘SRBM                               < 2e-16 ***
## ‘Missile Type‘Unknown                            4.72e-16 ***
## height                                           9.92e-07 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 36979.6  on 105  degrees of freedom
## Residual deviance:  6281.3  on  63  degrees of freedom
##   (98 observations deleted due to missingness)
## AIC: 7196.8
##
## Number of Fisher Scoring iterations: 5
```

Some of the key outputs of modelling can provide an indication of how accurate or useful it may be to make predictions. Very simply, the stars along the far right hand side indicate the parameter is significant in the model. We see many cases of three stars which is good. The AIC is 7196.8. Because the residual deviance is greater than the degrees of freedom, it indicates we may have over-dispersion in our model. Simply, this means the estimates are likely to be fairly good, but we’ll have broad error variance, and this is not ideal.

Prediction example

What we would do in proper modelling is work our way through a range of model diagnostics as well as use some libraries to determine via a range of methods the most effective choice of variables to use to predict the distance. However, this will become quite technical so we won't do that in this example. Rather, we'll show how we can use a model once we have tested that it is a good fit for the data and is consistent with our model assumptions.

In this simple case, we simply assign data to the three predictor variables (Facility Name, Missile Type and height) and observe the predicted distance.

```
# We'll pull out each variable so we can easily tweak them to see how it changes the output
"Facility Name" = "Pyongyang International Airport"
"Missile Type" = "SRBM"
height = 100
test.data = data.frame("Facility Name", "Missile Type", height)
predict(dist.glm, test.data, type = 'response')
```

```
##        1
## 518.3398
```

Some of the predictions seem mostly reasonable. However, be cautious to include a height for a missile type (like SRBM) that is far in excess of what we have in the dataset. Predictions in this case are not likely to be reliable. Predictions will tend to be most accurate when they are towards the centre of the values in the dataset, and least accurate when are they are beyond the ranges contained in the dataset.

Conclusion

This concludes the example data analysis on the North Korean missile test dataset. This example aimed to highlight the sorts of analyses that can be done with a curated *event* or *activity* dataset. This example focused on exploratory data analysis using primarily graphical analyses. Modelling and time series forecasting is a deeper more technical analysis that can be covered at a later stage.