# WSPR-Signal-Propagation-Analysis-HashemAlomari

June 6, 2024

WSPR-Signal-Propagation-Analysis

The code below each part of this question is related to that question and was used to answer it and/or contains the answers for that question.

Loading the wspr.csv data set into a data frame.

Are there any missing values?

```
Answer: yes there is one missing value located in the 'snr' column.
```

Performing any necessary data imputation on numerical data in the data frame…

I have imputated the row 100016 'snr' column value with the mean of the 'snr' column values.

```
In [3]: wspr <- read.csv('/public/bmort/R/wspr.csv')
        head(wspr, 5)
```

A data.frame: 5 Œ 20

| id | time | band | rx_sign | rx_lat | rx_lon | rx_loc | tx |
|---|---|---|---|---|---|---|---|
| <dbl> | <fct> | <int> | <fct> | <dbl> | <dbl> | <fct> | < |
| 6554621574 | 2023-11-01 00:00:00 | 10 | 01231 | 41.479 | 164.958 | RN21 | 0 |
| 6554625343 | 2023-11-01 00:00:00 | 10 | WB6JHI | 37.229 | -121.958 | CM97af | K |
| 6554625345 | 2023-11-01 00:00:00 | 10 | WB6JHI | 37.229 | -121.958 | CM97af | K |
| 6554625385 | 2023-11-01 00:00:00 | 10 | VE5CRL | 52.479 | -107.042 | DO62 | K |
| 6554625387 | 2023-11-01 00:00:00 | 10 | KD6EKQ | 32.854 | -117.208 | DM12ju | K |

```
In [34]: sum(is.na(wspr))
```

1

```
In [35]: colSums(is.na(wspr))
```

**id** 0 **time** 0 **band** 0 **rx_sign** 0 **rx_lat** 0 **rx_lon** 0 **rx_loc** 0 **tx_sign** 0 **tx_lat** 0 **tx_lon** 0 **tx_loc** 0 **distance** 0 **azimuth** 0 **rx_azimuth** 0 **frequency** 0 **power** 0 **snr** 1 **drift** 0 **version** 0 **code** 0

```
In [36]: mean(wspr$snr, na.rm = TRUE)
```

-14.7588258627529

```
In [37]: rows_with_na <- apply(is.na(wspr), 1, any)
         which(rows_with_na)
```

100016

```
In [38]: wspr[100016,'snr'] = mean(wspr$snr, na.rm = TRUE)
```

```
In [39]: wspr[100016,'snr']
```

-14.7588258627529
Lets analyze this data

i) How do the ranges of the values in the columns with numerical data compare?

   First of all, the range of the id column is over 2.4 million, which is very interesting because there are only 300 thousand rows, so it appears this dataset might be a subset of another or that the ids are inconsistent.

   The frequency column range has an exceptionally large range (over 2 billion), suggesting that the values in this column vary widely and might span several orders of magnitude. This could be indicative of very high-frequency measurements or a wide variety of frequencies recorded.

   The rx_lat.Range and tx_lat.Range show the range of latitudes, and rx_lon.Range and tx_lon.Range show the range of longitudes for the receiving (rx) and transmitting (tx) stations, respectively. Latitude ranges from -90 to 90, and longitude from -180 to 180, but the ranges are within a smaller subset of these values, suggesting limited geographical coverage.

   The azimuth.Range and rx_azimuth.Range have the same range of 359, which is one less than the full possible range of 360 degrees in a circle, suggesting that the azimuth measurements cover almost the full possible range.

   Others like power.Range and snr.Range have smaller, more moderate ranges. These columns likely represent more constrained measurements, such as power levels or quality of signal reception.

   When comparing the magnitude and ranges, it's clear that not all columns have similar magnitudes or ranges. Which is fine considering that this dataset contains different types of measurements.

ii) Does each column of numerical data have similar magnitudes and ranges?

   Each column has very different magnitudes and ranges, which is normal because some of the measured values have limited range, as mentioned in (i) when discussing longitudes, latitudes and azimuth. While other columns have extremely large values such as frequency and id.

iii) Are there any outliers? id time band rx_sign rx_lat rx_lon rx_loc TRUE NA TRUE NA TRUE TRUE NA tx_sign tx_lat tx_lon tx_loc distance azimuth rx_azimuth NA TRUE TRUE NA TRUE FALSE FALSE frequency power snr drift version code TRUE TRUE TRUE TRUE NA TRUE

   As we can see here, the only columns with no outliers are 'azituth' and 'rx_azimuth'

```
In [40]: summary(wspr)
```

```
      id                              time               band
 Min.   :6.555e+09   2023-11-01 00:28:00:  9352   Min.   :  -1.00
 1st Qu.:6.555e+09   2023-11-01 00:10:00:  8315   1st Qu.:   7.00
 Median :6.555e+09   2023-11-01 00:06:00:  8274   Median :  10.00
```

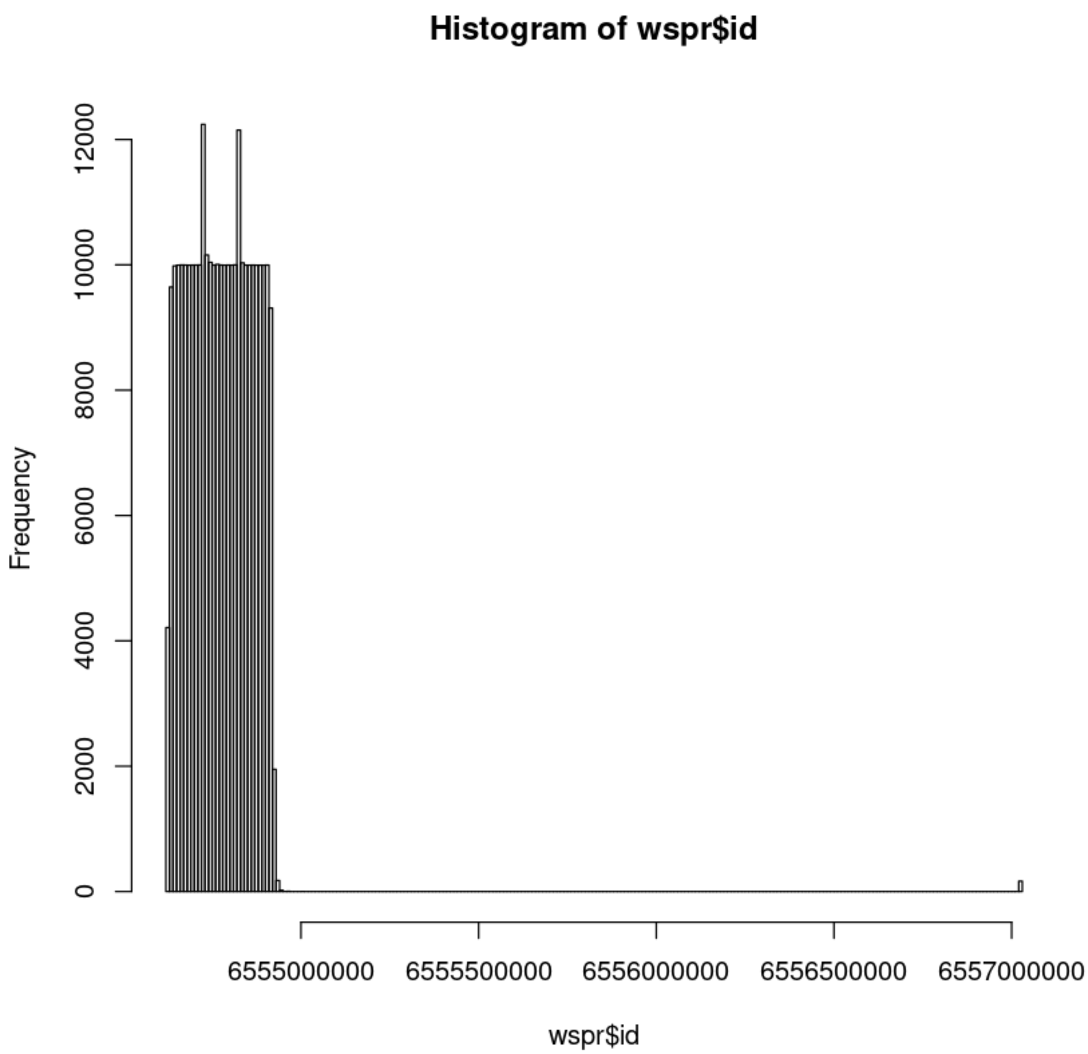
```
 Mean   :6.555e+09   2023-11-01 00:58:00:  8198   Mean   :  11.58  
 3rd Qu.:6.555e+09   2023-11-01 00:22:00:  8107   3rd Qu.:  14.00  
 Max.   :6.557e+09   2023-11-01 00:30:00:  7910   Max.   :2400.00  
                     (Other)            :249844                    
      rx_sign           rx_lat          rx_lon            rx_loc      
 KFS       :  4587   Min.   :-70.65   Min.   :-157.958   FN30lu :  6035  
 EA8/DF4UE:  4064   1st Qu.: 38.10   1st Qu.:-105.958   CM87tj :  4587  
 KA7OEI/Q :  3850   Median : 41.77   Median : -79.875   DN31uo :  4278  
 KPH       :  3753   Mean   : 41.07   Mean   : -60.402   IL38bp :  4064  
 WA2TP     :  3652   3rd Qu.: 47.69   3rd Qu.:   0.292   JN47wk :  3975  
 KK6PR     :  3428   Max.   : 68.35   Max.   : 175.875   CM88mc :  3753  
 (Other)   :276666                                       (Other):273308  
      tx_sign           tx_lat          tx_lon            tx_loc      
 TA4/G8SCU:  4858   Min.   :-87.52   Min.   :-173.042   KM56vo :  4861  
 WB6CXC    :  4696   1st Qu.: 34.73   1st Qu.: -99.042   FM19   :  4687  
 W8AC      :  3779   Median : 40.48   Median : -81.208   CM88lj :  4194  
 DP0POL    :  2919   Mean   : 39.73   Mean   : -62.202   EN91jm :  3779  
 NI5F      :  2833   3rd Qu.: 46.06   3rd Qu.:  -2.208   EM70fu :  2833  
 PA0MIK    :  2814   Max.   : 89.48   Max.   : 178.958   JO32mu :  2814  
 (Other)   :278101                                       (Other):276832  
    distance        azimuth        rx_azimuth      frequency        
 Min.   :    0   Min.   :  0.0   Min.   :  0.0   Min.   :1.374e+05  
 1st Qu.:  889   1st Qu.: 69.0   1st Qu.: 84.0   1st Qu.:7.040e+06  
 Median : 1584   Median :177.0   Median :180.0   Median :1.014e+07  
 Mean   : 2275   Mean   :177.5   Mean   :179.8   Mean   :1.176e+07  
 3rd Qu.: 2917   3rd Qu.:286.0   3rd Qu.:279.0   3rd Qu.:1.410e+07  
 Max.   :18925   Max.   :359.0   Max.   :359.0   Max.   :2.400e+09  
                                                                    
     power            snr             drift               version      
 Min.   :  0.0   Min.   :-40.00   Min.   :-4.00000   WD_3.0.9 : 39729  
 1st Qu.: 23.0   1st Qu.:-22.00   1st Qu.: 0.00000   2.6.1    : 36199  
 Median : 23.0   Median :-16.00   Median : 0.00000   WD_3.1.2 : 31032  
 Mean   : 26.5   Mean   :-14.76   Mean   :-0.05889            : 27038  
 3rd Qu.: 33.0   3rd Qu.: -9.00   3rd Qu.: 0.00000   1.4A Kiwi: 25850  
 Max.   :103.0   Max.   : 42.00   Max.   : 4.00000   2.5.4    : 17909  
                                                     (Other)  :122243  
      code       
 Min.   :-1.000  
 1st Qu.: 1.000  
 Median : 1.000  
 Mean   : 1.045  
 3rd Qu.: 1.000  
 Max.   : 8.000  
```

In [41]:
```
sapply(wspr, function(x) if(is.numeric(x)) c(Range = diff(range(x, na.rm = TRUE))) els
```

**id.Range** '2402141' **time** 'NOT APPLICABLE' **band.Range** '2401' **rx_sign** 'NOT

APPLICABLE' **rx_lat.Range** '139' **rx_lon.Range** '333.833' **rx_loc** 'NOT APPLICABLE' **tx_sign** 'NOT APPLICABLE' **tx_lat.Range** '177' **tx_lon.Range** '352' **tx_loc** 'NOT APPLICABLE' **distance.Range** '18925' **azimuth.Range** '359' **rx_azimuth.Range** '359' **frequency.Range** '2399932191' **power.Range** '103' **snr.Range** '82' **drift.Range** '8' **version** 'NOT APPLICABLE' **code.Range** '9'

```
In [42]: hist(wspr$id, breaks=sqrt(length(wspr$id))/2)
```




```
In [43]: has_outliers <- function(x) {
           if(is.numeric(x)) {
             Q1 <- quantile(x, 0.25, na.rm = TRUE)
             Q3 <- quantile(x, 0.75, na.rm = TRUE)
             IQR <- Q3 - Q1
```

```
        lower_bound <- Q1 - 1.5 * IQR
        upper_bound <- Q3 + 1.5 * IQR
        return(any(x < lower_bound | x > upper_bound))
      } else {
        return(NA) # Non-numeric columns cannot have numeric outliers, so we'll return NA
      }
    }

    sapply(wspr, has_outliers)
```

**id** TRUE **time** <NA> **band** TRUE **rx_sign** <NA> **rx_lat** TRUE **rx_lon** TRUE **rx_loc** <NA> **tx_sign** <NA> **tx_lat** TRUE **tx_lon** TRUE **tx_loc** <NA> **distance** TRUE **azimuth** FALSE **rx_azimuth** FALSE **frequency** TRUE **power** TRUE **snr** TRUE **drift** TRUE **version** <NA> **code** TRUE

C. How many unique values are in each of the following columns: band, rx_sign, and tx_sign?
answer:
'band': 23
'rx_sign': 891
'tx_sign': 1070

```
In [44]: length(unique(wspr$band))
```

23

```
In [45]: length(unique(wspr$rx_sign))
```

891

```
In [46]: length(unique(wspr$tx_sign))
```

1070

I will find some specific cases to get more familiar with the data: What is the average distance (in km) between the transmitting station and the receiving station for signals that have a power less than 30 dBm?
answer:
mean = 2057.33472735626

```
In [47]: distance_subset <- wspr[wspr$power < 30, 'distance']
         mean(distance_subset)
```

2057.33472735626

E. What is the call sign (rx_sign) of the receiving station that received the most signal transmissions on the 14 MHz band (i.e. band = 14)?
answer:
'EA8/DF4UE'

```
In [48]: band_subset <- (wspr[wspr$band == 14, 'rx_sign'])

In [49]: length(band_subset)
```

73667

In [51]: #unique(band_subset)

In [52]: length(unique(band_subset))

438

```
In [53]: tab <- tabulate(match(band_subset, unique(band_subset)))
         tab
```

1. 45 2. 627 3. 237 4. 120 5. 460 6. 8 7. 7 8. 119 9. 13 10. 48 11. 240 12. 112 13. 285 14. 123 15. 16 16. 218 17. 60 18. 282 19. 48 20. 382 21. 67 22. 80 23. 612 24. 12 25. 210 26. 7 27. 84 28. 19 29. 588 30. 626 31. 157 32. 143 33. 324 34. 228 35. 609 36. 635 37. 361 38. 71 39. 217 40. 48 41. 116 42. 611 43. 301 44. 535 45. 749 46. 626 47. 7 48. 717 49. 138 50. 147 51. 157 52. 32 53. 1123 54. 120 55. 84 56. 96 57. 577 58. 2 59. 177 60. 616 61. 99 62. 817 63. 90 64. 209 65. 14 66. 46 67. 18 68. 5 69. 133 70. 444 71. 12 72. 109 73. 111 74. 55 75. 659 76. 266 77. 53 78. 258 79. 229 80. 35 81. 491 82. 445 83. 115 84. 445 85. 97 86. 199 87. 6 88. 346 89. 68 90. 310 91. 11 92. 241 93. 17 94. 69 95. 39 96. 44 97. 458 98. 205 99. 746 100. 136 101. 142 102. 11 103. 129 104. 110 105. 362 106. 109 107. 364 108. 288 109. 595 110. 75 111. 23 112. 323 113. 36 114. 537 115. 416 116. 592 117. 89 118. 7 119. 47 120. 351 121. 387 122. 17 123. 437 124. 14 125. 311 126. 728 127. 476 128. 11 129. 586 130. 45 131. 55 132. 395 133. 44 134. 424 135. 22 136. 415 137. 523 138. 346 139. 955 140. 142 141. 260 142. 622 143. 31 144. 357 145. 113 146. 316 147. 132 148. 52 149. 48 150. 436 151. 37 152. 222 153. 13 154. 855 155. 865 156. 1354 157. 1012 158. 1054 159. 711 160. 884 161. 237 162. 8 163. 69 164. 81 165. 118 166. 633 167. 77 168. 220 169. 690 170. 801 171. 545 172. 48 173. 41 174. 892 175. 22 176. 1170 177. 1337 178. 549 179. 775 180. 321 181. 574 182. 943 183. 312 184. 17 185. 112 186. 102 187. 928 188. 967 189. 65 190. 54 191. 178 192. 319 193. 59 194. 260 195. 66 196. 130 197. 65 198. 197 199. 356 200. 169 201. 292 202. 33 203. 249 204. 35 205. 249 206. 28 207. 18 208. 121 209. 268 210. 226 211. 267 212. 318 213. 28 214. 91 215. 584 216. 48 217. 372 218. 910 219. 461 220. 92 221. 112 222. 199 223. 267 224. 9 225. 59 226. 30 227. 105 228. 25 229. 32 230. 54 231. 23 232. 18 233. 217 234. 11 235. 31 236. 9 237. 20 238. 47 239. 28 240. 32 241. 272 242. 96 243. 106 244. 59 245. 189 246. 517 247. 28 248. 57 249. 30 250. 162 251. 294 252. 9 253. 14 254. 9 255. 306 256. 17 257. 87 258. 57 259. 130 260. 2 261. 4 262. 38 263. 24 264. 84 265. 23 266. 17 267. 31 268. 164 269. 15 270. 32 271. 4 272. 16 273. 14 274. 16 275. 204 276. 98 277. 24 278. 17 279. 16 280. 241 281. 71 282. 40 283. 45 284. 51 285. 32 286. 8 287. 689 288. 4 289. 4 290. 8 291. 2 292. 3 293. 7 294. 4 295. 4 296. 24 297. 29 298. 21 299. 4 300. 3 301. 8 302. 5 303. 3 304. 11 305. 4 306. 183 307. 125 308. 42 309. 4 310. 9 311. 5 312. 9 313. 3 314. 4 315. 11 316. 4 317. 5 318. 23 319. 7 320. 4 321. 25 322. 8 323. 38 324. 34 325. 6 326. 4 327. 15 328. 28 329. 4 330. 41 331. 2 332. 2 333. 188 334. 5 335. 1 336. 36 337. 71 338. 36 339. 35 340. 2 341. 23 342. 14 343. 4 344. 99 345. 83 346. 22 347. 5 348. 73 349. 18 350. 5 351. 3 352. 65 353. 17 354. 289 355. 60 356. 1 357. 70 358. 15 359. 29 360. 77 361. 76 362. 8 363. 23 364. 69 365. 133 366. 9 367. 9 368. 120 369. 139 370. 66 371. 18 372. 21 373. 18 374. 5 375. 60 376. 19 377. 10 378. 54 379. 77 380. 152 381. 30 382. 24 383. 16 384. 23 385. 25 386. 171 387. 14 388. 7 389. 349 390. 8 391. 25 392. 4 393. 3 394. 6 395. 5 396. 9 397. 63 398. 9 399. 16 400. 1 401. 7 402. 7 403. 223 404. 2 405. 2 406. 1 407. 15 408. 34 409. 2 410. 4 411. 7 412. 38 413. 1 414. 2 415. 242 416. 1 417. 1 418. 10 419. 2 420. 2 421. 1 422. 4 423. 51 424. 33 425. 2 426. 1 427. 1 428. 1 429. 1 430. 1 431. 1 432. 1 433. 2 434. 1 435. 1 436. 1 437. 1 438. 1

```
In [54]: most_freq <- max(tab)
         most_freq
```

1354

```
In [55]: mode_values <- unique(band_subset)[tab == most_freq]
         mode_values
```

EA8/DF4UE *Levels*: 1. '01230' 2. '01231' 3. '14OA159' 4. '161OS344' 5. '2E0DSS' 6. '2E0ILY/KIWI' 7. '2E0PYB' 8. '2E0TMI' 9. '2E0VOE-SDR' 10. '2E0ZPM' 11. '2M0HZD' 12. '3XFRN0' 13. '56IGJF' 14. '5B4AOE' 15. '5B4CY' 16. '5Q9T' 17. '7L4IOU' 18. '7L4IOU2' 19. '9A3KB' 20. '9K2RA' 21. '9V1KG' 22. 'AA4IH' 23. 'AA7NM' 24. 'AA8DT' 25. 'AB1BZ' 26. 'AB1KW' 27. 'AB1YX' 28. 'AC0G' 29. 'AC1BC' 30. 'AC1IM' 31. 'AC3V' 32. 'AC3V/3' 33. 'AC7IJ' 34. 'AC9RZ' 35. 'AD0X' 36. 'AE2EA' 37. 'AE4WX' 38. 'AF5GM' 39. 'AI2CW' 40. 'AI4Y/S' 41. 'AI6VN/KH6' 42. 'AJ8S' 43. 'AK4T' 44. 'AL7CR' 45. 'AL7DS' 46. 'AX37EUDXF' 47. 'BA4XX' 48. 'BEDFORD' 49. 'BM2KVV' 50. 'BM4AIK' 51. 'BM7GUP' 52. 'BUZZARD' 53. 'BV2YD' 54. 'BV3UN' 55. 'BV5ON' 56. 'BV7AU' 57. 'BV7YA' 58. 'BX2AJC' 59. 'BX4ACP' 60. 'BX6ABC' 61. 'CE3BSK' 62. 'CE3VRT' 63. 'CT1EBQ' 64. 'CT1ETL/1' 65. 'DA5UDI' 66. 'DB1UJ-1' 67. 'DB9OH' 68. 'DC0DX' 69. 'DC1RDB' 70. 'DC5AL-R' 71. 'DC7JZB/P' 72. 'DC7JZB/RX' 73. 'DC7TO' 74. 'DD1NK' 75. 'DD5XX' 76. 'DD8HL' 77. 'DE1LON' 78. 'DF1DR' 79. 'DF1QQ/SDR' 80. 'DF1VB' 81. 'DF2JP' 82. 'DF2MG' 83. 'DF6NM' 84. 'DF8FH' 85. 'DF8JO' 86. 'DF8OE' 87. 'DG5SEK' 88. 'DG7RJ' 89. 'DJ1GT' 90. 'DJ2MG' 91. 'DJ5MHZ' 92. 'DJ6DK' 93. 'DJ9BS' 94. 'DK0ABT' 95. 'DK2DB' 96. 'DK4AA' 97. 'DK5HH' 98. 'DK5HH/3' 99. 'DK6UG' 100. 'DK7FC' 101. 'DK8FT' 102. 'DK8FT/A' 103. 'DK8NE' 104. 'DL0AO' 105. 'DL0DTM' 106. 'DL0HOT/E' 107. 'DL0HOT/H' 108. 'DL0HT' 109. 'DL0LV' 110. 'DL0PF' 111. 'DL1HMJ' 112. 'DL1MF' 113. 'DL1SER' 114. 'DL2BBC' 115. 'DL2JA' 116. 'DL2ZZ' 117. 'DL3EL' 118. 'DL3ROB' 119. 'DL4AD' 120. 'DL4EAI' 121. 'DL4KBB' 122. 'DL4RAJ' 123. 'DL4RU' 124. 'DL4SA' 125. 'DL4TOM/RX' 126. 'DL4XJ' 127. 'DL4ZBE' 128. 'DL5ALW' 129. 'DL6EBP' 130. 'DL6PLZ' 131. 'DL7AUQ' 132. 'DL8FCL' 133. 'DL8SEL' 134. 'DL8SFE' 135. 'DL9GTB' 136. 'DO1EHS' 137. 'DO3RCX' 138. 'DO5EU' 139. 'DO8OT' 140. 'DP0GVN' 141. 'DP0GVN/1' 142. 'DP0GVN/3' 143. 'DP0POL' 144. 'E28AC' 145. 'EA3IHV' 146. 'EA3IHV/KIWI' 147. 'EA5JEG' 148. 'EA5ZL-1' 149. 'EA7GF' 150. 'EA8/DF4UE' 151. 'EA8BFK' 152. 'EA8BVP' 153. 'EA8CHC' 154. 'EC5M' 155. 'EI3JCB' 156. 'EI4ACB' 157. 'EI7CLB' 158. 'EI7GL' 159. 'EI7II' 160. 'EW8MKU' 161. 'EW8RD' 162. 'F1EYG' 163. 'F1HDI' 164. 'F1JSC' 165. 'F1UBL' 166. 'F1VAM' 167. 'F1VMV' 168. 'F4FPR' 169. 'F4IDD' 170. 'F4KJI' 171. 'F4KKF' 172. 'F4VTQ' 173. 'F4VTQ/A' 174. 'F5178SWL' 175. 'F5MMX' 176. 'F6KBF' 177. 'F6PRA' 178. 'F8COD' 179. 'FN41RY' 180. 'G0API' 181. 'G0FCA' 182. 'G0FCH' 183. 'G0FYD' 184. 'G0GHK/SDR' 185. 'G0HIU' 186. 'G0IDE' 187. 'G0IFI' 188. 'G0KTN' 189. 'G0LUJ' 190. 'G0LUJ/SDR1' 191. 'G0MQW' 192. 'G0PYB' 193. 'G0VCA' 194. 'G3PWJ/KIWI' 195. 'G3VPW' 196. 'G3WCB' 197. 'G3XBM' 198. 'G3ZYD' 199. 'G4BOO' 200. 'G4BRK' 201. 'G4CLO' 202. 'G4CUI' 203. 'G4FBA' 204. 'G4GCI' 205. 'G4LCM' 206. 'G4MEM/K' 207. 'G4MSA' 208. 'G4SDL' 209. 'G4SZM-1' 210. 'G4TNU' 211. 'G4ZFQ' 212. 'G6AVK' 213. 'G6GN' 214. 'G6JTB' 215. 'G6KSN' 216. 'G6MC/K' 217. 'G6OLU' 218. 'G6UQZ' 219. 'G6UZT' 220. 'G7RHF' 221. 'G8AOE' 222. 'G8CQX' 223. 'G8LCO' 224. 'G8NXD' 225. 'G8ORM' 226. 'G8SEZ' 227. 'GB0SNB/SDR' 228. 'GM0UDL' 229. 'GM1OXB' 230. 'GM3RQQ' 231. 'GM3YXM' 232. 'GM4DTH' 233. 'GW4JJW' 234. 'HA3MG' 235. 'HA3PG' 236. 'HA3PMF' 237. 'HA5UDS' 238. 'HAXX.SPACE' 239. 'HB0SM' 240. 'HB4SF' 241. 'HB9ASB' 242. 'HB9BXU' 243. 'HB9DSE' 244. 'HB9FX' 245. 'HB9HJZ' 246. 'HB9ODP' 247. 'HB9PAE' 248. 'HB9TCS' 249. 'HB9TMC' 250. 'HB9TTU' 251. 'HB9VQQ' 252. 'HB9VQQ/K' 253. 'HL3AMO-1' 254. 'HL3AMO-2' 255. 'HP1COO' 256. 'I0UVN' 257. 'I55387FI' 258. 'I72000/VE' 259. 'INUVIK' 260. 'IQ2BG' 261. 'IU1QQM' 262. 'IU3NTY' 263. 'IU5MQQ' 264. 'IU7RAL' 265. 'IV3IHD' 266. 'IW2DMO' 267. 'IW3HBX' 268. 'IZ6QQT' 269. 'IZ7SLZ' 270. 'J41PAN' 271. 'JA1KPF' 272. 'JA1NQI'

273. 'JA5FFO' 274. 'JA5YLT' 275. 'JA7KBR' 276. 'JE1JDL' 277. 'JG1PIX' 278. 'JG6DMH' 279. 'JH1ARY' 280. 'JH3KTL' 281. 'JJ8NTM' 282. 'JK1LOT' 283. 'JM1CZS' 284. 'JM8FLR/SDR' 285. 'JN1ZOG' 286. 'JP1HUJ' 287. 'JP1ODJ/SDR' 288. 'JQ1ZJU' 289. 'JS1YDE' 290. 'JS1YEG' 291. 'K0BL' 292. 'K0RE' 293. 'K1BZ' 294. 'K1ENB' 295. 'K1HTV-4' 296. 'K1MEA' 297. 'K1RA' 298. 'K1RA-PI' 299. 'K1VL' 300. 'K1YZY' 301. 'K2JY' 302. 'K3RWR' 303. 'K3XR' 304. 'K4BYN' 305. 'K4CLE' 306. 'K4COD' 307. 'K4IQJ/4' 308. 'K4PQC' 309. 'K4RCA' 310. 'K4SBB' 311. 'K4UAH' 312. 'K5CGM' 313. 'K5EM' 314. 'K5KHK' 315. 'K5MO-1' 316. 'K5PK' 317. 'K5SWA' 318. 'K5XL' 319. 'K6FOD' 320. 'K6JFZ' 321. 'K6PZB' 322. 'K6RFT' 323. 'K6RRP' 324. 'K7API' 325. 'K7BIZ' 326. 'K7FET' 327. 'K7GXB' 328. 'K7MAP' 329. 'K7MDL' 330. 'K7VNZ' 331. 'K8CHY' 332. 'K8RGC' 333. 'K8TV' 334. 'K9AN' 335. 'K9DZT' 336. 'K9IMM' 337. 'K9KFR' 338. 'K9PAW' 339. 'K9REO' 340. 'K9SWX' 341. 'K9VD' 342. 'K9XT' 343. 'K9YWO' 344. 'KA0EQP' 345. 'KA0FCT' 346. 'KA1R' 347. 'KA3BWP' 348. 'KA4PKB/0' 349. 'KA7OEI' 350. 'KA7OEI-1' 351. 'KA7OEI/Q' 352. 'KA8HUZ' 353. 'KA9LHE' 354. 'KA9Q' 355. 'KB0LQJ' 356. 'KB0VYG' 357. 'KB4NEW' 358. 'KB7GF' 359. 'KB7YVW' 360. 'KB8VME' 361. 'KC0KVR' 362. 'KC0MYY' 363. 'KC1RID' 364. 'KC2LSB' 365. 'KC2STA1' 366. 'KC2STA4' 367. 'KC3FL' 368. 'KC3IBR' 369. 'KC3PQG' 370. 'KC5NK' 371. 'KC5YOD' 372. 'KC6FLG' 373. 'KC6HAI' 374. 'KC6WPK' 375. 'KC9VNU' 376. 'KC9YRA' 377. 'KD0BQS' 378. 'KD0HFC' 379. 'KD0LKI' 380. 'KD0PXX' 381. 'KD2CLR' 382. 'KD2LZI' 383. 'KD2OM' 384. 'KD2SGY' 385. 'KD2VDN' 386. 'KD2YGN' 387. 'KD5J' 388. 'KD6EKQ' 389. 'KD6FTR' 390. 'KD6RF' 391. 'KD7HGL' 392. 'KD7UHR' 393. 'KD9HOE' 394. 'KD9KHZ' 395. 'KD9LCN' 396. 'KD9NYE' 397. 'KD9PAH' 398. 'KD9QZO' 399. 'KE4IP' 400. 'KE5HPY' 401. 'KE5ZBG' 402. 'KE6GG' 403. 'KE7A' 404. 'KE7HR' 405. 'KE8MYP' 406. 'KE8Y' 407. 'KE8ZCS' 408. 'KF0MBA' 409. 'KF4EAG' 410. 'KF4HCW' 411. 'KF4JU' 412. 'KF4LIJ' 413. 'KF4ZNO' 414. 'KF5GCF' 415. 'KF6ZEO' 416. 'KF7O' 417. 'KFS' 418. 'KG5ABO' 419. 'KG5FNU' 420. 'KG5GNQ' 421. 'KG6NFJ' 422. 'KG7YC' 423. 'KG9IV' 424. 'KH6FA' 425. 'KI8FW' 426. 'KJ6EI/VE7' 427. 'KJ6MKI' 428. 'KJ6MKI/6' 429. 'KJ7ZGC' 430. 'KK1D' 431. 'KK4UNE' 432. 'KK6AYK' 433. 'KK6EEW' 434. 'KK6PR' 435. 'KL3RR' 436. 'KL7L' 437. 'KL7QT-7' 438. 'KM3T-2' 439. 'KM3T-3' 440. 'KM4BLG/SWL' 441. 'KM4RK' 442. 'KM5SW' 443. 'KM5SW/SE' 444. 'KM6CQ' 445. 'KM6MWO' 446. 'KM6WXB' 447. 'KN4JTB' 448. 'KN4NBI' 449. 'KN4TGQ' 450. 'KN6KSF' 451. 'KN6NK' 452. 'KN6SVA' 453. 'KN6WVP' 454. 'KO4HRW' 455. 'KO4WSU' 456. 'KP4MD' 457. 'KPH' 458. 'KQ2Y' 459. 'KQ6RS/6' 460. 'KR6LA' 461. 'KR7O' 462. 'KRSEPRZ' 463. 'KV0S' 464. 'KV4TT' 465. 'KV4XY' 466. 'KV6X' 467. 'KX4AZ/T' 468. 'KX4AZ/W' 469. 'LA/DL1HWK' 470. 'LA1ITS' 471. 'LA1PLA' 472. 'LA1ZM' 473. 'LA3EQ' 474. 'LA3FY/2' 475. 'LA4CIA' 476. 'LB0K' 477. 'LB3AH' 478. 'LU3DJ' 479. 'LW6DLS' 480. 'LW6EQG' 481. 'LX1DQ' 482. 'M0HGU' 483. 'M0IQF' 484. 'M0LLP' 485. 'M0LYN' 486. 'M0PWX' 487. 'M0TLJ' 488. 'M0UNI' 489. 'M0VSE' 490. 'M0WHU' 491. 'M0XDK' 492. 'M1DST' 493. 'M6IXY' 494. 'M7MMP' 495. 'MM3NDH' 496. 'MW0MUT' 497. 'MW7SIF' 498. 'N0GQ' 499. 'N0QBH' 500. 'N1CL' 501. 'N1DQ-HL' 502. 'N1PCE' 503. 'N1SDP' 504. 'N1SER' 505. 'N1SRC/L' 506. 'N1VF/1' 507. 'N2HQI' 508. 'N2YCH' 509. 'N3AGE' 510. 'N3EDS' 511. 'N3HZZ' 512. 'N3LSB' 513. 'N3PK' 514. 'N4BUT' 515. 'N4JJS' 516. 'N4PRQ' 517. 'N4TKW' 518. 'N4TTN' 519. 'N4TVC/4' 520. 'N4WLO' 521. 'N5AQM' 522. 'N5ATR' 523. 'N5LXN' 524. 'N5ODX' 525. 'N5PYK' 526. 'N5UTV' 527. 'N5XXN' 528. 'N6GN/K' 529. 'N6GN2' 530. 'N6IO/K' 531. 'N6KOG' 532. 'N6RY' 533. 'N7VVX' 534. 'N8AYY' 535. 'N8BB' 536. 'N8DAW' 537. 'N8DFC' 538. 'N8GA' 539. 'N8OBJ' 540. 'N8OWY' 541. 'N8VIM' 542. 'N8XMV' 543. 'N9LB' 544. 'N9MKC' 545. 'N9RU' 546. 'N9YBX' 547. 'ND7M' 548. 'NG1Y' 549. 'NH6V' 550. 'NH6XO' 551. 'NI0L' 552. 'NI5F' 553. 'NM7A' 554. 'NN0V' 555. 'NS8C' 556. 'NV4X' 557. 'NW0C' 558. 'OE3GBB' 559. 'OE3GBB/Q' 560. 'OE3XHU' 561. 'OE6ADD' 562. 'OE6ADD/6' 563. 'OE6CUD' 564. 'OE6PWD' 565. 'OE9GHV' 566. 'OE9GHV/Q' 567. 'OE9HLH' 568. 'OE9TAV' 569. 'OH1LSQ' 570. 'OH3FR' 571. 'OH6AH'

572. ’OH6BG’ 573. ’OH6GAP’ 574. ’OH8GKP’ 575. ’OH8XAT’ 576. ’OK1008SWL’ 577. ’OK1HRA’ 578. ’OK1RAJ’ 579. ’OK2BVG’ 580. ’OK2IP’ 581. ’ON4LAO’ 582. ’ON4RST’ 583. ’ON4WS/RX’ 584. ’ON5KQ’ 585. ’OZ1KVB’ 586. ’OZ2JBR’ 587. ’OZ4JJ’ 588. ’OZ7IT’ 589. ’OZ9ACF’ 590. ’PA0LSB’ 591. ’PA0LSB/L’ 592. ’PA0LSB/V’ 593. ’PA0MLC’ 594. ’PA0RDT’ 595. ’PA0SLT/4’ 596. ’PA0SLT/A’ 597. ’PA1JMS’ 598. ’PA1W/P’ 599. ’PA3ABK’ 600. ’PA3BTI’ 601. ’PA3BUL’ 602. ’PA3FNY’ 603. ’PA3FNY2’ 604. ’PA3FNY3’ 605. ’PA3FNY4’ 606. ’PA3GSH’ 607. ’PA3HEA’ 608. ’PA3WLE’ 609. ’PA4HJH’ 610. ’PA7EY/2’ 611. ’PA7EY/5’ 612. ’PA7EY/6’ 613. ’PA8E’ 614. ’PC9DB’ 615. ’PD0DNL’ 616. ’PD0OHW’ 617. ’PD1V’ 618. ’PD2BAS’ 619. ’PD7SAS’ 620. ’PE1EZC’ 621. ’PE1INM’ 622. ’PH4RTM’ 623. ’PHONEMARX’ 624. ’PI4THT’ 625. ’PP5ZX’ 626. ’PU5ALE’ 627. ’PY2GN’ 628. ’PY2SDR’ 629. ’R0CDO’ 630. ’R1LB/RX’ 631. ’R2BAS’ 632. ’RA0SMS’ 633. ’RA4FHE’ 634. ’RC7SWL’ 635. ’RJ35’ 636. ’RU6CH’ 637. ’RX9O’ 638. ’RZ3DVP’ 639. ’S51RS’ 640. ’S54MI’ 641. ’SA2LLL’ 642. ’SA6BSS/RP’ 643. ’SA7LIL’ 644. ’SDR-JN69BA’ 645. ’SM0EPX/RX2’ 646. ’SM2KOT’ 647. ’SM2LKW’ 648. ’SM3LNM’ 649. ’SM4JST/RX1’ 650. ’SM5-8574’ 651. ’SM5FGQ’ 652. ’SM7KHA’ 653. ’SV1EAI’ 654. ’SV8/SV1SJJ’ 655. ’SV8FET’ 656. ’SV8RV’ 657. ’SWL-FN03HS35’ 658. ’SWL-FRANK’ 659. ’SWL/EWE’ 660. ’SWL/HU1UB’ 661. ’SWL13RHX’ 662. ’SWLDM63ID’ 663. ’SWLKCL’ 664. ’SWLSPM001’ 665. ’SWLSREY0123’ 666. ’TA4CQ’ 667. ’TF3HZ’ 668. ’TF4AH’ 669. ’TF4M’ 670. ’TI4JWC’ 671. ’TOMA1564’ 672. ’UB1AKX’ 673. ’UB3AQI’ 674. ’UNLIS’ 675. ’UR6LQP’ 676. ’VA2ZAC’ 677. ’VA3DP’ 678. ’VA3IRF’ 679. ’VA3ROM’ 680. ’VA3WBA’ 681. ’VA6SMH’ 682. ’VA7GEM’ 683. ’VA7JX’ 684. ’VA7MM’ 685. ’VE1INN’ 686. ’VE2BJG’ 687. ’VE2GAV’ 688. ’VE2SUS’ 689. ’VE2UG’ 690. ’VE3CUB’ 691. ’VE3CWM’ 692. ’VE3DE’ 693. ’VE3EBR’ 694. ’VE3GEN’ 695. ’VE3GTC’ 696. ’VE3MNA’ 697. ’VE3PRO’ 698. ’VE3TAZ’ 699. ’VE3XLT’ 700. ’VE4BJZ’ 701. ’VE4CA’ 702. ’VE4FEB’ 703. ’VE4KRK’ 704. ’VE5CRL’ 705. ’VE5HRC’ 706. ’VE6EGN’ 707. ’VE6JY’ 708. ’VE6PDQ’ 709. ’VE7AFV’ 710. ’VE7AV’ 711. ’VE7BDQ’ 712. ’VE7IRN’ 713. ’VE7RVS’ 714. ’VE7WEX’ 715. ’VE7YIO’ 716. ’VE9SWL’ 717. ’VI2AZ’ 718. ’VK1KW’ 719. ’VK2BLS’ 720. ’VK2DAY’ 721. ’VK2DVM’ 722. ’VK2IJM’ 723. ’VK2JMR’ 724. ’VK2KCM’ 725. ’VK2MAX’ 726. ’VK2TLQ’ 727. ’VK2TMC’ 728. ’VK2VU’ 729. ’VK2XDG’ 730. ’VK2XGJ’ 731. ’VK2ZEE’ 732. ’VK3DXE’ 733. ’VK3FC’ 734. ’VK3KEZ’ 735. ’VK3KHZ’ 736. ’VK3MAP’ 737. ’VK3TPM’ 738. ’VK3YFD’ 739. ’VK3ZTE’ 740. ’VK4BAP’ 741. ’VK4EMM’ 742. ’VK4QW’ 743. ’VK4XDB’ 744. ’VK5ADE’ 745. ’VK5AJQ’ 746. ’VK5ARG’ 747. ’VK5ATN/A’ 748. ’VK5FQ/W’ 749. ’VK5HW’ 750. ’VK5HW-1’ 751. ’VK5ZBI’ 752. ’VK6RWS’ 753. ’VK6TB’ 754. ’VK6WR’ 755. ’VK7AM’ 756. ’VK7AP’ 757. ’VK7CMV’ 758. ’VK7DC’ 759. ’VK7DIK’ 760. ’VK7EKA’ 761. ’VK7JJ’ 762. ’VK7MO’ 763. ’VK7PD’ 764. ’VK7TW’ 765. ’VK7ZAB’ 766. ’VK7ZLB’ 767. ’VR2BG’ 768. ’VR2FUN-74’ 769. ’VR2KJ’ 770. ’W0ADL’ 771. ’W0AIR’ 772. ’W0ASW’ 773. ’W0AY’ 774. ’W0IOO’ 775. ’W0QV’ 776. ’W0VI’ 777. ’W0YSE’ 778. ’W1/AJ8S’ 779. ’W1BW-1’ 780. ’W1CK’ 781. ’W1CLM’ 782. ’W1FRV’ 783. ’W1VB’ 784. ’W1VD’ 785. ’W2ACR’ 786. ’W2GSA’ 787. ’W3ABW’ 788. ’W3BX’ 789. ’W3ENR’ 790. ’W3HH’ 791. ’W3LLA’ 792. ’W3SKY’ 793. ’W3TS’ 794. ’W4BXA’ 795. ’W4DES’ 796. ’W4HOD’ 797. ’W4KEL’ 798. ’W4SSC’ 799. ’W4WLO’ 800. ’W5GFI’ 801. ’W5WTH’ 802. ’W6ABL’ 803. ’W6EXT’ 804. ’W6LVP’ 805. ’W6NJX’ 806. ’W6TQS’ 807. ’W7BOT’ 808. ’W7DG’ 809. ’W7EME’ 810. ’W7PUA’ 811. ’W7STF’ 812. ’W7WKR’ 813. ’W7WKR/K’ 814. ’W7WND’ 815. ’W7YSB’ 816. ’W8AC’ 817. ’W8JI’ 818. ’W8UM’ 819. ’W9XA’ 820. ’WA0R’ 821. ’WA1RAJ’ 822. ’WA2EUJ’ 823. ’WA2QZP’ 824. ’WA2TP’ 825. ’WA2TP/Q’ 826. ’WA2ZKD’ 827. ’WA3BM’ 828. ’WA3DNM’ 829. ’WA3IAC’ 830. ’WA3NAN’ 831. ’WA3TTS’ 832. ’WA3U’ 833. ’WA4ZZW’ 834. ’WA5DJJ’ 835. ’WA5DXP’ 836. ’WA6JRW’ 837. ’WA9FIO’ 838. ’WA9NWW’ 839. ’WA9WTK’ 840. ’WB0KWJ’ 841. ’WB1BQE’ 842. ’WB2EEE’ 843. ’WB3AVN’ 844. ’WB5B’ 845. ’WB5SRK’ 846. ’WB6JHI’ 847. ’WB6YAZ’ 848. ’WB7ABP/K’ 849. ’WB7ABP/Q’ 850. ’WB8DQT’ 851. ’WB8ILI’ 852. ’WB8RJY’ 853. ’WC2L’ 854. ’WD0E’ 855. ’WD4ELG’ 856. ’WD8QLY’ 857. ’WD9EKA’ 858. ’WEBSDRGK’ 859. ’WESSEXSDR’ 860. ’WESTONSDR’

861. 'WF7U' 862. 'WF7W' 863. 'WI6P' 864. 'WO2S' 865. 'WS3W' 866. 'WV5L' 867. 'WW6D' 868. 'WY1U' 869. 'WY7JMS' 870. 'YL3ANT' 871. 'YO3WL' 872. 'YO7COV' 873. 'YO8STB' 874. 'YU1ADO' 875. 'YY7ECA' 876. 'ZL1CE' 877. 'ZL1KFM' 878. 'ZL1TJK' 879. 'ZL2005SWL' 880. 'ZL2BCI' 881. 'ZL2RKL' 882. 'ZL3TKI' 883. 'ZL4CAT' 884. 'ZL4KYH' 885. 'ZS3Y/SDR-CW' 886. 'ZS3Y/SDR-DL' 887. 'ZS3Y/SDR-DLV' 888. 'ZS3Y/SDR-SD' 889. 'ZS5RTL' 890. 'ZS6BQQ' 891. 'ZS6JGL'

Now I will partition the WSPR data set so that a random sample of 80% of the data will be used for training and 20% will be used for testing your machine learning model.

```
In [56]: library(caret)
```

```
In [57]: training_size <- floor(0.80 * nrow(wspr))
         training_indices <- sample(seq_len(nrow(wspr)), size = training_size)
         training_set <- wspr[training_indices, ]

         testing_set <- wspr[-training_indices, ]
```

Now I will generate a linear regression model to predict the signal-to-noise ratio from the distance, frequency, and power.

```
In [58]: model_formula <- snr ~ distance + frequency + power

         #Training the Model
         linear_model <- lm(model_formula, data = training_set) # Use your training dataset he

         #Summary of the Model
         summary(linear_model)
```

```
Call:
lm(formula = model_formula, data = training_set)

Residuals:
    Min      1Q  Median      3Q     Max
-34.879  -6.374  -0.894   5.475  59.252

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.026e+01  6.726e-02 -301.25  < 2e-16 ***
distance    -1.073e-03  8.223e-06 -130.51  < 2e-16 ***
frequency    1.353e-09  2.987e-10    4.53 5.91e-06 ***
power        2.988e-01  2.442e-03  122.37  < 2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 8.662 on 239996 degrees of freedom
Multiple R-squared:  0.1049,Adjusted R-squared:  0.1049
F-statistic:  9376 on 3 and 239996 DF,  p-value: < 2.2e-16
```

I will Use the 20% of the data set aside for testing to determine the accuracy of the model. Choosing an appropriate accuracy metric.

I will be answering the following:

i) How well does the model predict the signal to noise values from distance, frequency, and power?

ii) Why is the accuracy is good or poor

In [59]:
```
predictions <- predict(linear_model, newdata = testing_set)

actual_snr <- testing_set$snr

mse <- mean((actual_snr - predictions)^2)

rmse <- sqrt(mse)

print(paste("MSE:", mse))
print(paste("RMSE:", rmse))
```

```
[1] "MSE: 74.9414594482498"
[1] "RMSE: 8.65687353772999"
```

i) answer:

The RMSE here lets us know that that the model's errors are somewhat high, given that the RMSE is more than half of the mean SNR value. This could be problematic in a practical sense because, in the context of radio signal propagation, such deviations might lead to incorrect conclusions or expectations about signal quality.

With the mean SNR at around 14, and considering the nature of the data and the importance of accurate predictions for SNR, the accuracy of the model might not be as good as needed. If the SNR values typically do not vary widely (e.g., if they usually range from 10 to 20), an RMSE of 8.66 could be too large relative to this range.

ii) answer:

Given the complexity of atmospheric conditions affecting signal propagation, it's possible that linear regression might not be able to capture all the necessary patterns and might be too simplistic a model. Many other factors included in the dataset that are not just power distance and frequency can introduce a lot of variability into the SNR.

Variables including Azimuth, time, drift, and the geographical locations and almost all the other features in the data set would help make better predictions. Using only 3 will make sure that the values are most accurate.

It is also worth considering whether there is a non-linear relationship between the predictors and the SNR or if there are interactions between predictors that a linear model does not account for. In such cases, a different modeling approach, like polynomial regression or a non-linear model, might yield more accurate predictions.

I. What is the predicted signal to noise value for a receiver that is located 2,000 km from a transmitter that uses a frequency of 14,030,000 Hz and a power level of 31 dBm? How confident are you in the answer? Explain your reasoning.

answer: snr = -13.1301119878796

I'm not very confident in this answer, the features used to predict this are very limited since we are only using 3 features. Going back to the error we see that this model has very high error when predicting signal-to-noise based on power, frequency and distance.

However, what makes me a little more confident about this is that the values of the features seem to be within the ranges of the original dataset, which increases the reliability of the predictions.

```
In [60]: new_data <- data.frame(distance = 2000, frequency = 14030000, power = 31)

         predict(linear_model, newdata = new_data)
```

**1:** -13.1249313142831