DAT320: Compulsory assignment 1

Group 4

2024-09-16

```
options(contrasts = c("contr.sum", "contr.poly"))
require("ggplot2")
require("dplyr")
require("ppcor")
require("caret")
require("tidyr")
```

Exercise 1 - R syntax & data structures

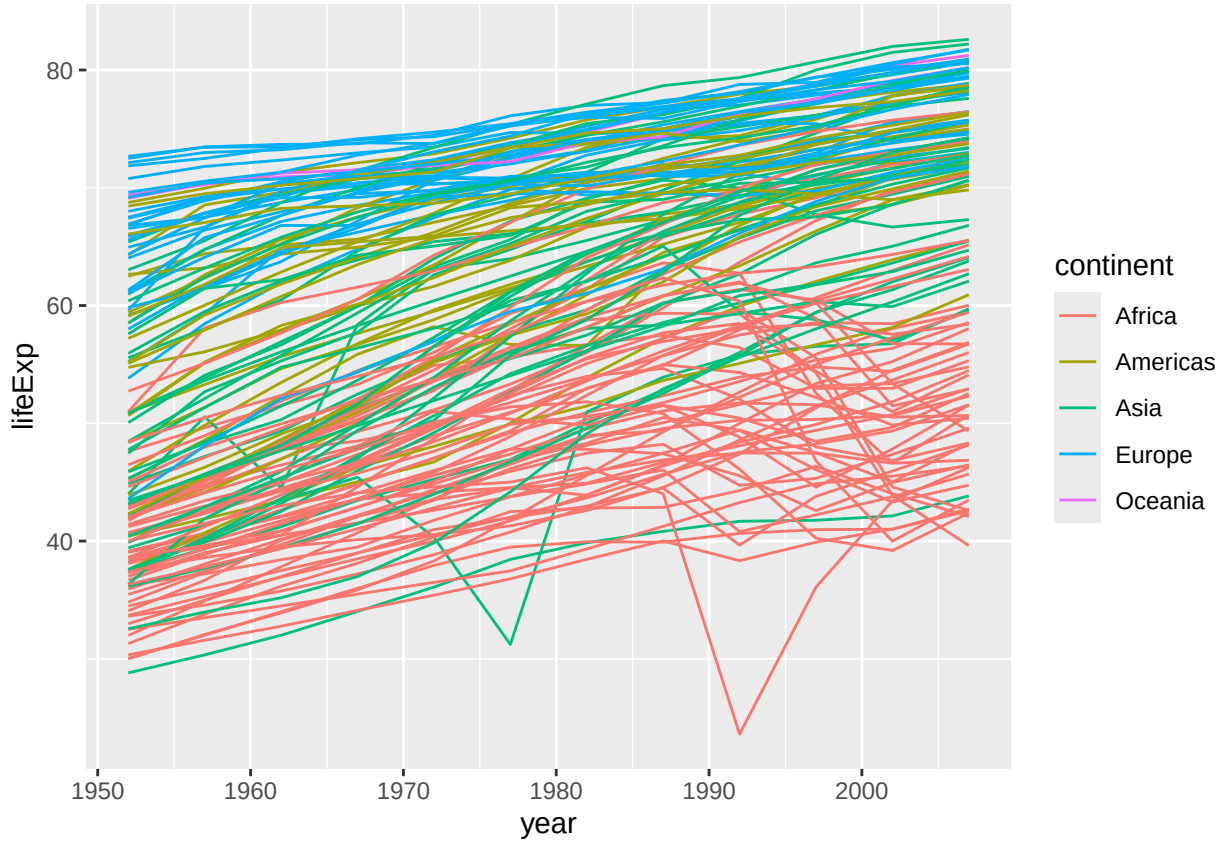
Task a

```
gapminder <- read.csv("gapminder.csv")

summary(gapminder)
```

```
##        X            country           continent              year     
##  Min.   :   1.0   Length:1704        Length:1704        Min.   :1952  
##  1st Qu.: 426.8   Class :character   Class :character   1st Qu.:1966  
##  Median : 852.5   Mode  :character   Mode  :character   Median :1980  
##  Mean   : 852.5                                        Mean   :1980  
##  3rd Qu.:1278.2                                        3rd Qu.:1993  
##  Max.   :1704.0                                        Max.   :2007  
##     lifeExp           pop             gdpPercap       
##  Min.   :23.60   Min.   :6.001e+04   Min.   :   241.2  
##  1st Qu.:48.20   1st Qu.:2.794e+06   1st Qu.:  1202.1  
##  Median :60.71   Median :7.024e+06   Median :  3531.8  
##  Mean   :59.47   Mean   :2.960e+07   Mean   :  7215.3  
##  3rd Qu.:70.85   3rd Qu.:1.959e+07   3rd Qu.:  9325.5  
##  Max.   :82.60   Max.   :1.319e+09   Max.   :113523.1
```

Task b

```
gapminder %>%
  group_by(country) %>%
  ggplot(aes(x=year, y=lifeExp, gorup=country,  colour=continent)) +
    geom_line()
```



Task c

```
knitr::kable(
  gapminder %>%
    group_by(continent, year) %>%
    summarise_at(vars(lifeExp), list(Min = min, Med = median, Mean = mean, Max = max, Sd = sd)) %>%
    data.frame()
)
```

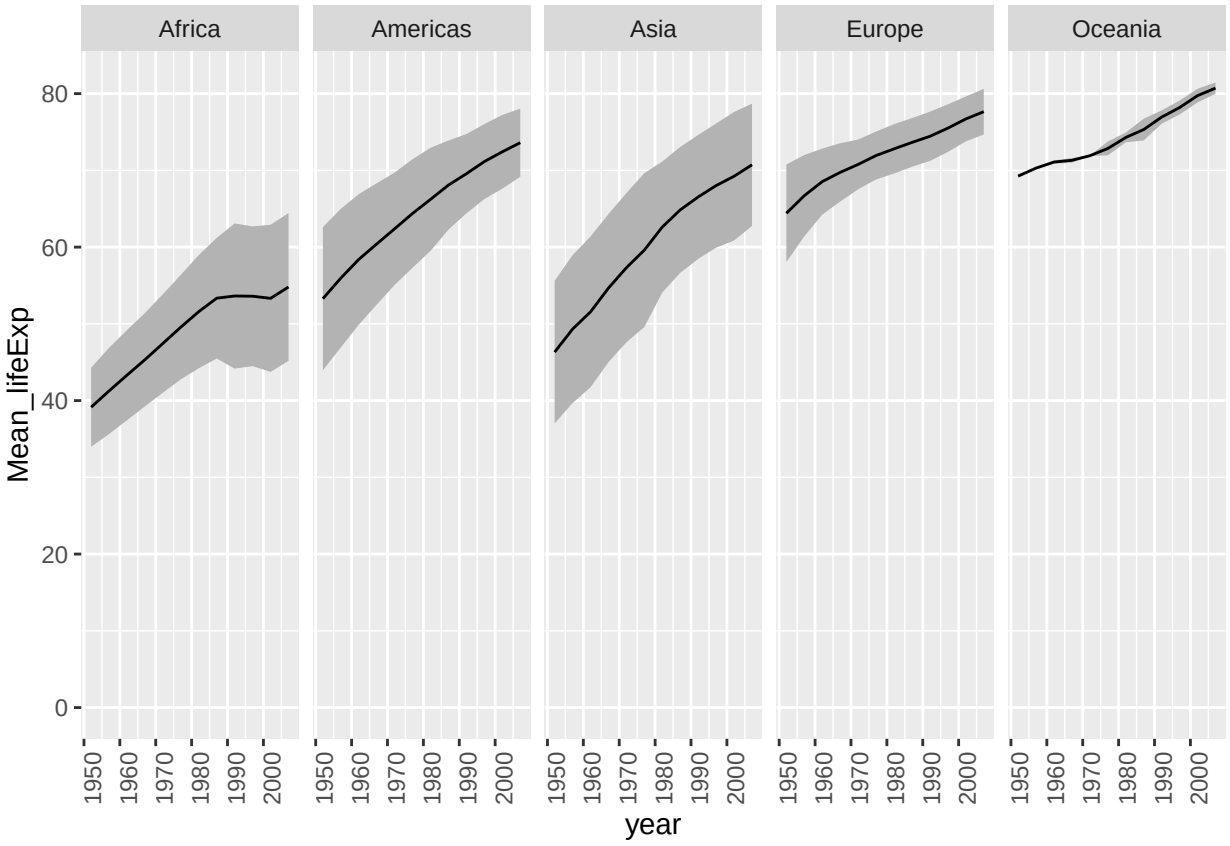
continent	year	Min	Med	Mean	Max	Sd
Africa	1952	30.000	38.8330	39.13550	52.724	5.1515814
Africa	1957	31.570	40.5925	41.26635	58.089	5.6201229
Africa	1962	32.767	42.6305	43.31944	60.246	5.8753639
Africa	1967	34.113	44.6985	45.33454	61.557	6.0826726
Africa	1972	35.400	47.0315	47.45094	64.274	6.4162583
Africa	1977	36.788	49.2725	49.58042	67.064	6.8081974
Africa	1982	38.445	50.7560	51.59287	69.885	7.3759401
Africa	1987	39.906	51.6395	53.34479	71.913	7.8640891
Africa	1992	23.599	52.4290	53.62958	73.615	9.4610710
Africa	1997	36.087	52.7590	53.59827	74.772	9.1033866
Africa	2002	39.193	51.2355	53.32523	75.744	9.5864959
Africa	2007	39.613	52.9265	54.80604	76.442	9.6307807
Americas	1952	37.579	54.7450	53.27984	68.750	9.3260819
Americas	1957	40.696	56.0740	55.96028	69.960	9.0331923
Americas	1962	43.428	58.2990	58.39876	71.300	8.5035437

continent	year	Min	Med	Mean	Max	Sd
Americas	1967	45.032	60.5230	60.41092	72.130	7.9091710
Americas	1972	46.714	63.4410	62.39492	72.880	7.3230168
Americas	1977	49.923	66.3530	64.39156	74.210	7.0694956
Americas	1982	51.461	67.4050	66.22884	75.760	6.7208338
Americas	1987	53.636	69.4980	68.09072	76.860	5.8019288
Americas	1992	55.089	69.8620	69.56836	77.950	5.1671038
Americas	1997	56.671	72.1460	71.15048	78.610	4.8875839
Americas	2002	58.137	72.0470	72.42204	79.770	4.7997055
Americas	2007	60.916	72.8990	73.60812	80.653	4.4409476
Asia	1952	28.801	44.8690	46.31439	65.390	9.2917507
Asia	1957	30.332	48.2840	49.31854	67.840	9.6354286
Asia	1962	31.997	49.3250	51.56322	69.390	9.8206319
Asia	1967	34.020	53.6550	54.66364	71.430	9.6509646
Asia	1972	36.088	56.9500	57.31927	73.420	9.7227000
Asia	1977	31.220	60.7650	59.61056	75.380	10.0221970
Asia	1982	39.854	63.7390	62.61794	77.110	8.5352214
Asia	1987	40.822	66.2950	64.85118	78.670	8.2037919
Asia	1992	41.674	68.6900	66.53721	79.360	8.0755490
Asia	1997	41.763	70.2650	68.02052	80.690	8.0911706
Asia	2002	42.129	71.0280	69.23388	82.000	8.3745954
Asia	2007	43.828	72.3960	70.72848	82.603	7.9637245
Europe	1952	43.585	65.9000	64.40850	72.670	6.3610883
Europe	1957	48.079	67.6500	66.70307	73.470	5.2958054
Europe	1962	52.098	69.5250	68.53923	73.680	4.3024996
Europe	1967	54.336	70.6100	69.73760	74.160	3.7997285
Europe	1972	57.005	70.8850	70.77503	74.720	3.2405764
Europe	1977	59.507	72.3350	71.93777	76.110	3.1210300
Europe	1982	61.036	73.4900	72.80640	76.990	3.2182603
Europe	1987	63.108	74.8150	73.64217	77.410	3.1696803
Europe	1992	66.146	75.4510	74.44010	78.770	3.2097811
Europe	1997	68.835	76.1160	75.50517	79.390	3.1046766
Europe	2002	70.845	77.5365	76.70060	80.620	2.9221796
Europe	2007	71.777	78.6085	77.64860	81.757	2.9798127
Oceania	1952	69.120	69.2550	69.25500	69.390	0.1909188
Oceania	1957	70.260	70.2950	70.29500	70.330	0.0494975
Oceania	1962	70.930	71.0850	71.08500	71.240	0.2192031
Oceania	1967	71.100	71.3100	71.31000	71.520	0.2969848
Oceania	1972	71.890	71.9100	71.91000	71.930	0.0282843
Oceania	1977	72.220	72.8550	72.85500	73.490	0.8980256
Oceania	1982	73.840	74.2900	74.29000	74.740	0.6363961
Oceania	1987	74.320	75.3200	75.32000	76.320	1.4142136
Oceania	1992	76.330	76.9450	76.94500	77.560	0.8697413
Oceania	1997	77.550	78.1900	78.19000	78.830	0.9050967
Oceania	2002	79.110	79.7400	79.74000	80.370	0.8909545
Oceania	2007	80.204	80.7195	80.71950	81.235	0.7290271

Task d

```
gapminder %>%
  group_by(continent, year) %>%
  summarise(Mean_lifeExp=mean(lifeExp, na.rm = T), SD_lifeExp=sd(lifeExp, na.rm = T), .groups = "drop")
```

```
ggplot(aes(x=year, y=Mean_lifeExp)) +
geom_ribbon(aes(ymin= Mean_lifeExp - SD_lifeExp, ymax = Mean_lifeExp + SD_lifeExp), fill = "grey70") +
geom_line() +
facet_grid(.~continent) +
theme(axis.text.x = element_text(angle=90)) +
ylim(0, NA)
```



Exercise 2 - Elementary data analysis and model training

Task a

```
weatherHistory <- read.csv("weatherHistory.csv")
head(weatherHistory)
```

```
##                  Formatted.Date       Summary Precip.Type Temperature..C.
## 1 2006-04-01 00:00:00.000 +0200 Partly Cloudy        rain        9.472222
## 2 2006-04-01 01:00:00.000 +0200 Partly Cloudy        rain        9.355556
## 3 2006-04-01 02:00:00.000 +0200 Mostly Cloudy        rain        9.377778
## 4 2006-04-01 03:00:00.000 +0200 Partly Cloudy        rain        8.288889
## 5 2006-04-01 04:00:00.000 +0200 Mostly Cloudy        rain        8.755556
## 6 2006-04-01 05:00:00.000 +0200 Partly Cloudy        rain        9.222222
##   Apparent.Temperature..C. Humidity Wind.Speed..km.h. Wind.Bearing..degrees.
## 1                 7.388889     0.89           14.1197                    251
## 2                 7.227778     0.86           14.2646                    259
## 3                 9.377778     0.89            3.9284                    204
```

```
## 4                 5.944444     0.83           14.1036                    269
## 5                 6.977778     0.83           11.0446                    259
## 6                 7.111111     0.85           13.9587                    258
##   Visibility..km. Loud.Cover Pressure..millibars.
## 1         15.8263          0              1015.13
## 2         15.8263          0              1015.63
## 3         14.9569          0              1015.94
## 4         15.8263          0              1016.41
## 5         15.8263          0              1016.51
## 6         14.9569          0              1016.66
##                       Daily.Summary
## 1 Partly cloudy throughout the day.
## 2 Partly cloudy throughout the day.
## 3 Partly cloudy throughout the day.
## 4 Partly cloudy throughout the day.
## 5 Partly cloudy throughout the day.
## 6 Partly cloudy throughout the day.
```

Qualitative nominal

- Summary
- Precip.Type
- Daily.Summary

Quantitative Continuous:

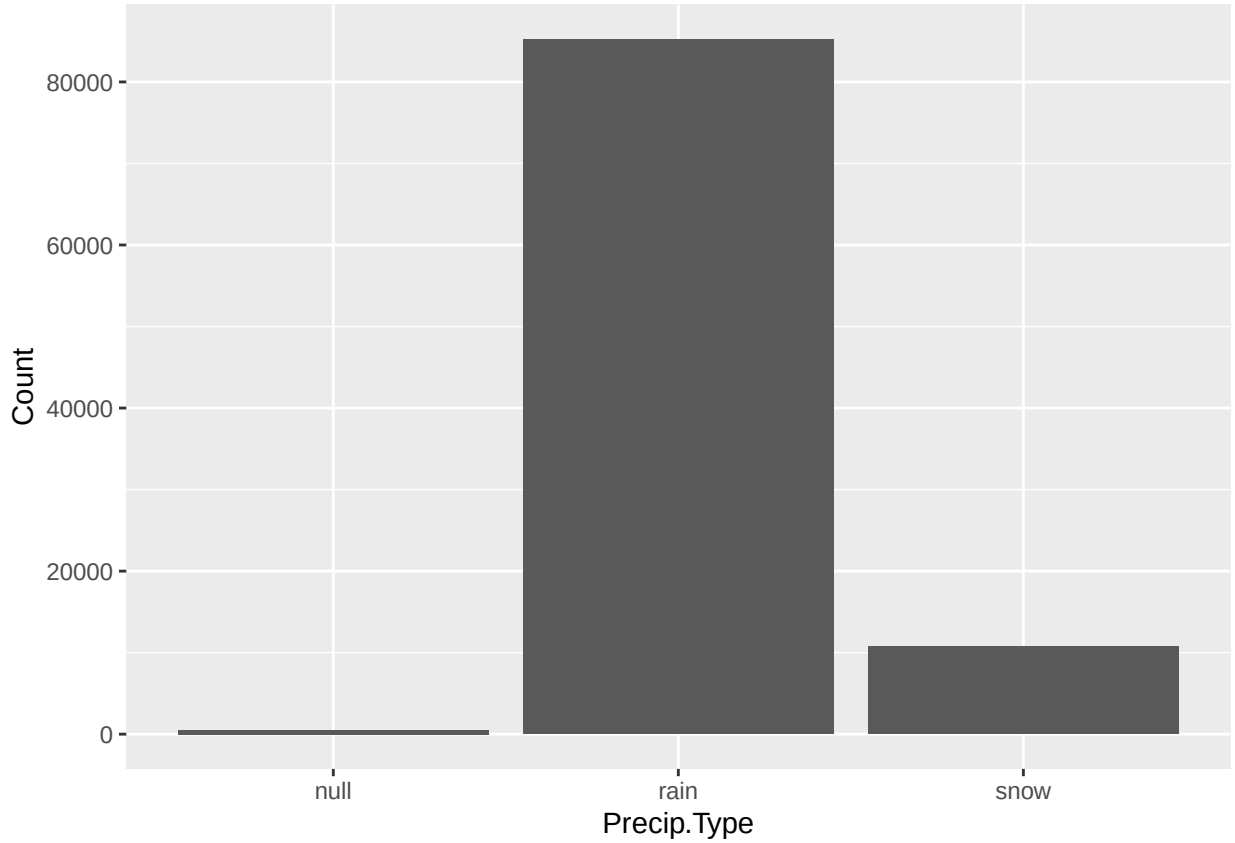
- Temperature..C.
- Apparent.Temperature..C.
- Humidity
- Wind.Speed..km.h.
- Visibility..km.
- Wind.Bearing..degrees (Reason: Not ranked)

Quantitative Discrete:

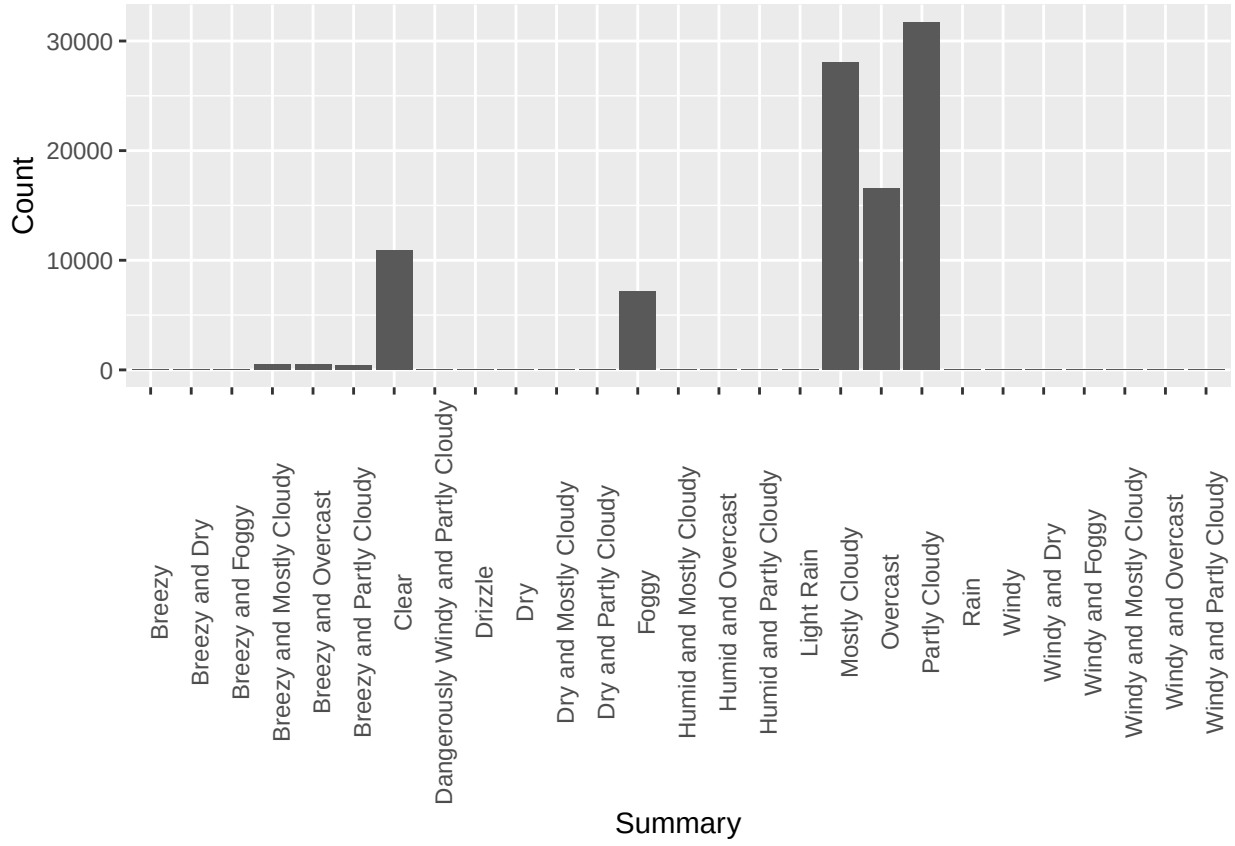
- Formatted.Date
- Loud.Cover

Qualitative nominal

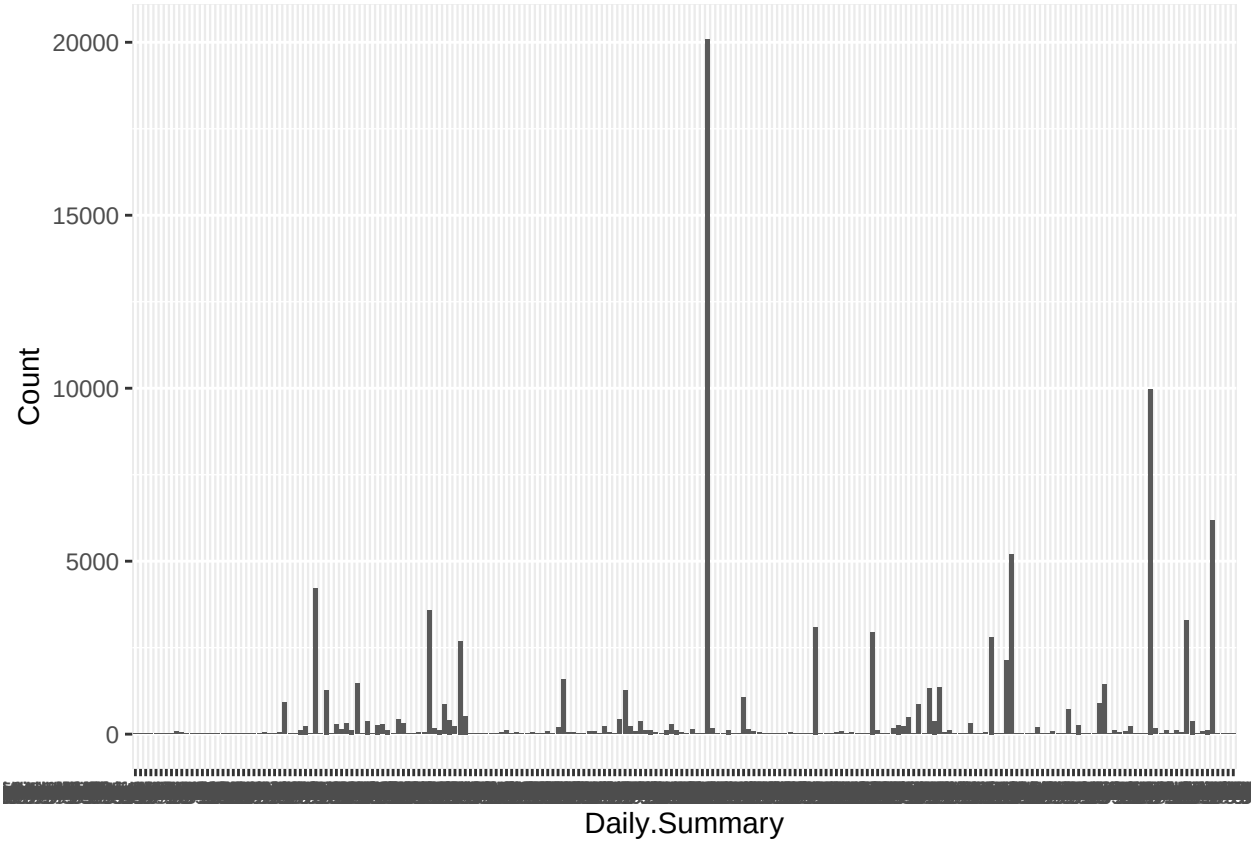
```
weatherHistory %>%
  group_by(Precip.Type) %>%
  summarize(Count = n()) %>%
  ggplot(aes(x=Precip.Type, y=Count)) +
  geom_bar(stat='identity', position='dodge')
```




```
weatherHistory %>%
  group_by(Summary) %>%
  summarize(Count = n()) %>%
  ggplot(aes(x=Summary, y=Count)) +
  geom_bar(stat='identity', position='dodge') +
  theme(axis.text.x = element_text(angle=90))
```




```
weatherHistory %>%
  group_by(Daily.Summary) %>%
  summarize(Count = n()) %>%
  ggplot(aes(x=Daily.Summary, y=Count)) +
  geom_bar(stat='identity', position='dodge')
```

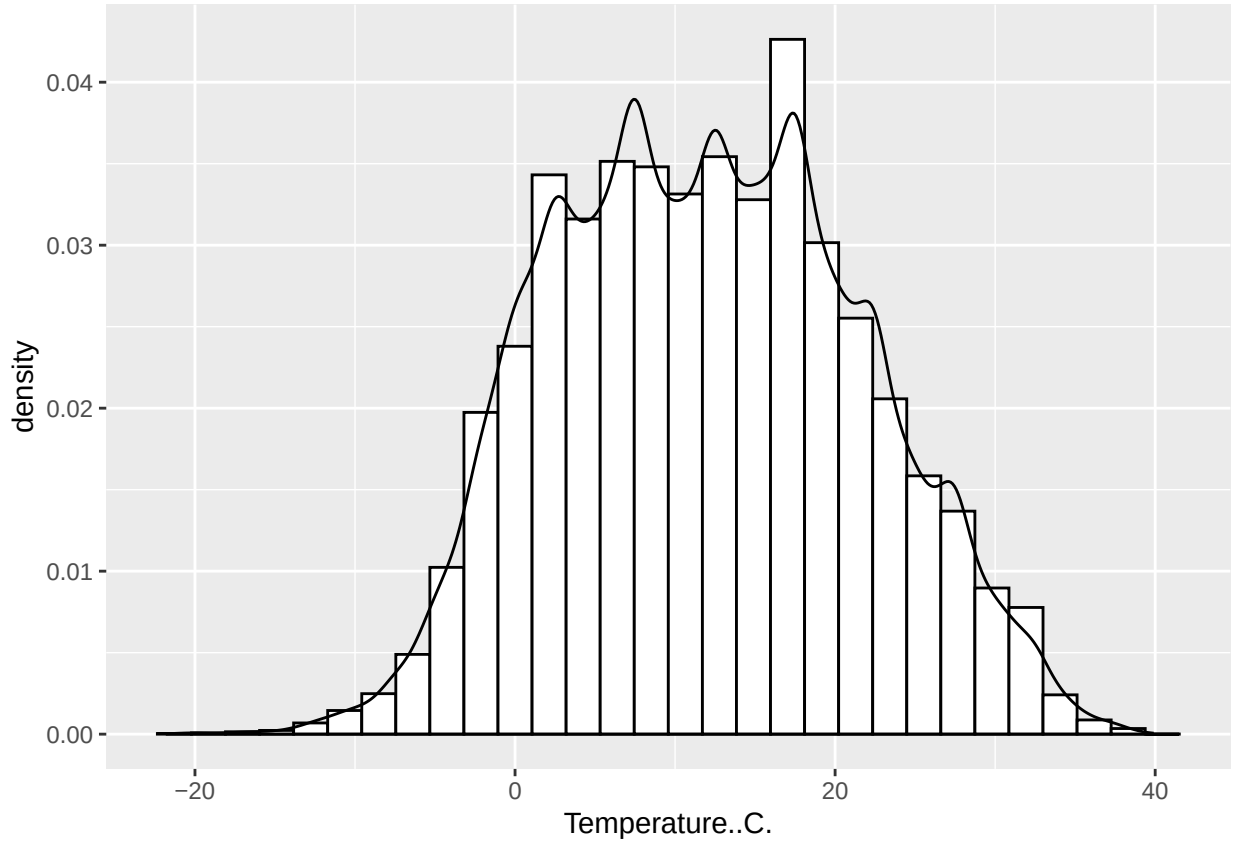


Discrete nominal

```
weatherHistory %>%
  ggplot(aes(Temperature..C.)) +
  geom_histogram(aes(y = ..density..), fill = "white", color="black") +
  stat_density(kernel = "gaussian", fill = NA, colour = "black")
```

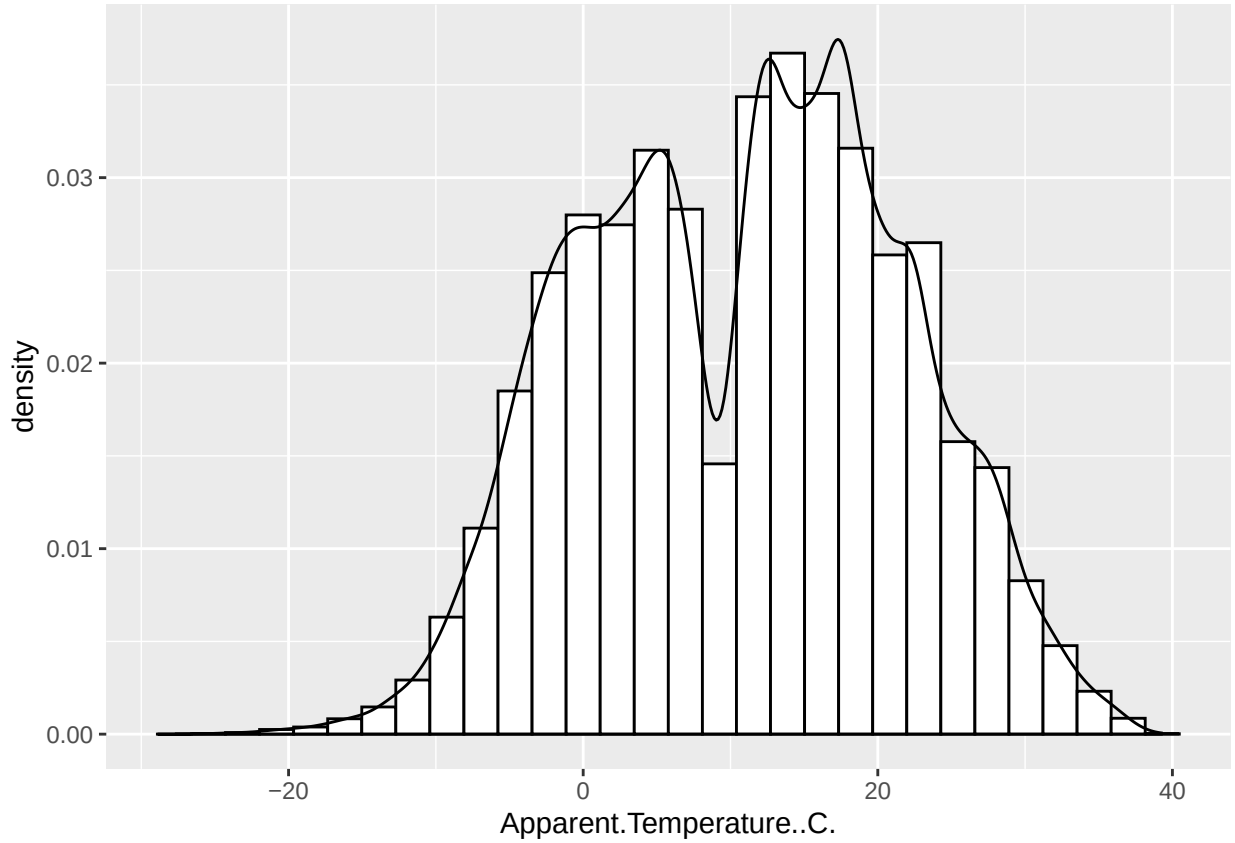
```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



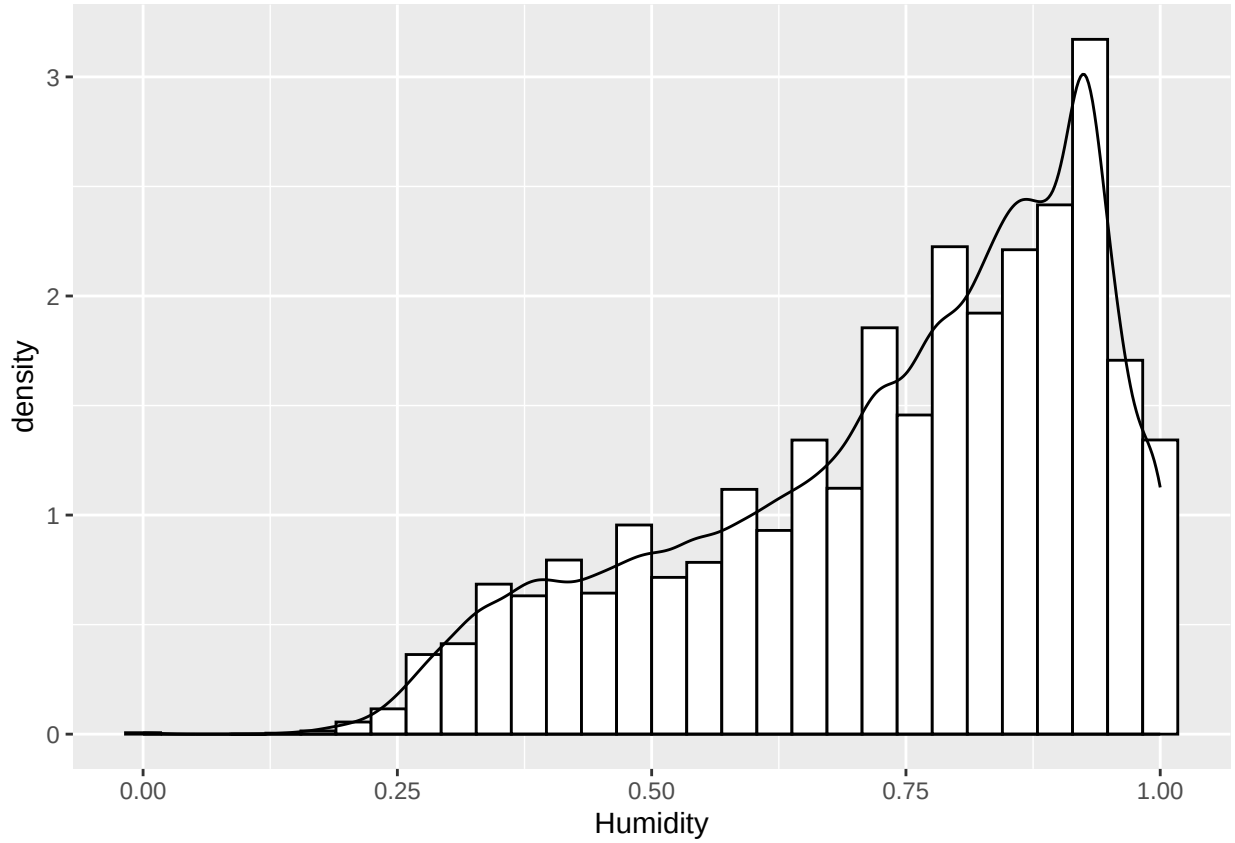

```
weatherHistory %>%
  ggplot(aes(Apparent.Temperature..C.)) +
  geom_histogram(aes(y = ..density..), fill = "white", color="black") +
  stat_density(kernel = "gaussian", fill = NA, colour = "black")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



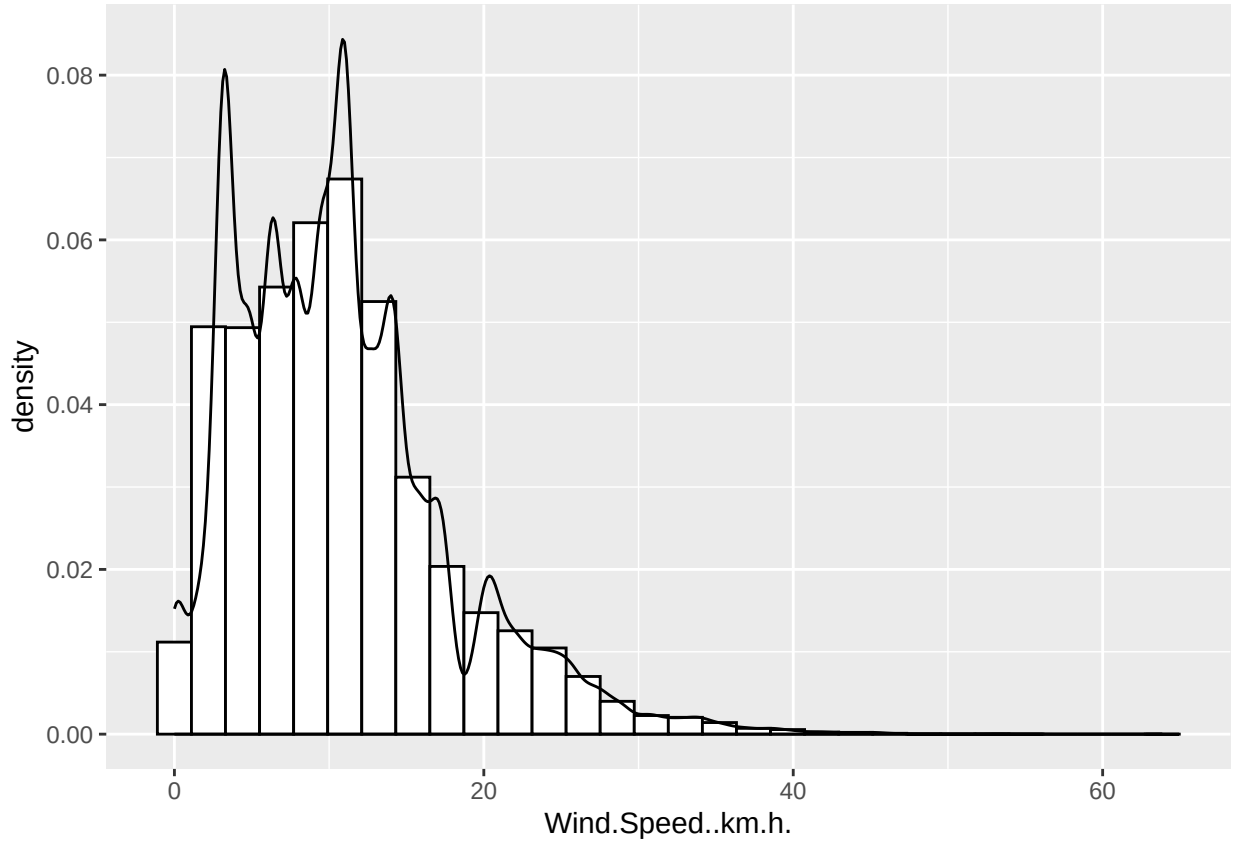

```
weatherHistory %>%
  ggplot(aes(Humidity)) +
  geom_histogram(aes(y = ..density..), fill = "white", color="black") +
  stat_density(kernel = "gaussian", fill = NA, colour = "black")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



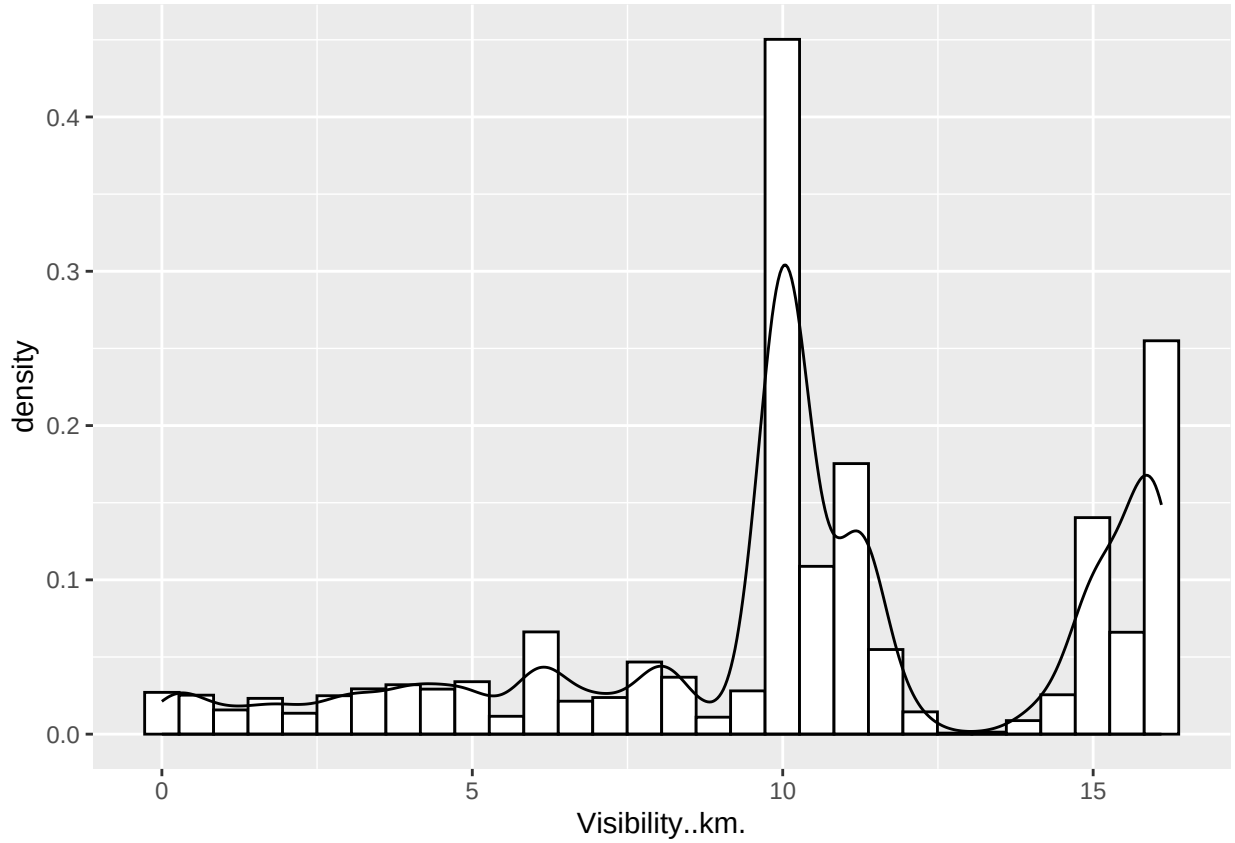

```
weatherHistory %>%
  ggplot(aes(Wind.Speed..km.h.)) +
  geom_histogram(aes(y = ..density..), fill = "white", color="black") +
  stat_density(kernel = "gaussian", fill = NA, colour = "black")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



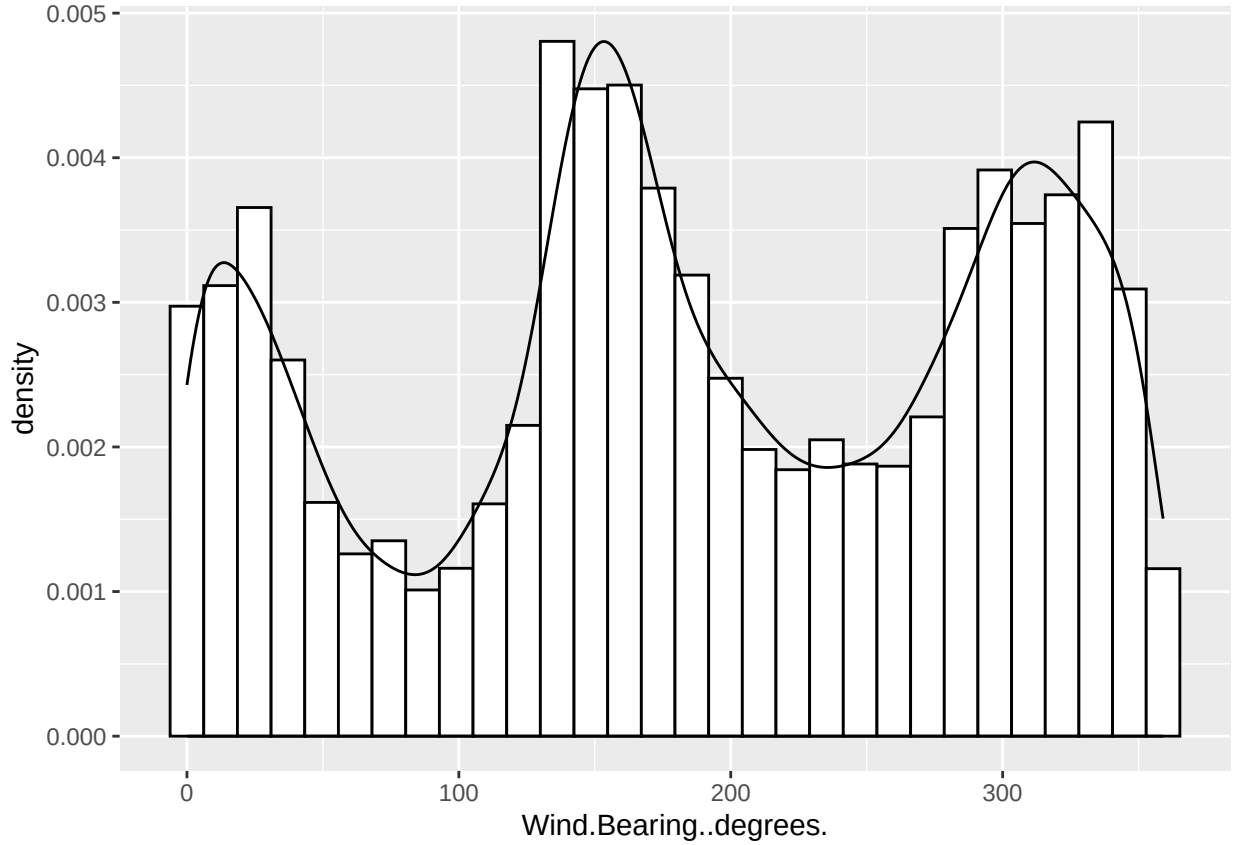

```
weatherHistory %>%
  ggplot(aes(Visibility..km.)) +
  geom_histogram(aes(y = ..density..), fill = "white", color="black") +
  stat_density(kernel = "gaussian", fill = NA, colour = "black")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



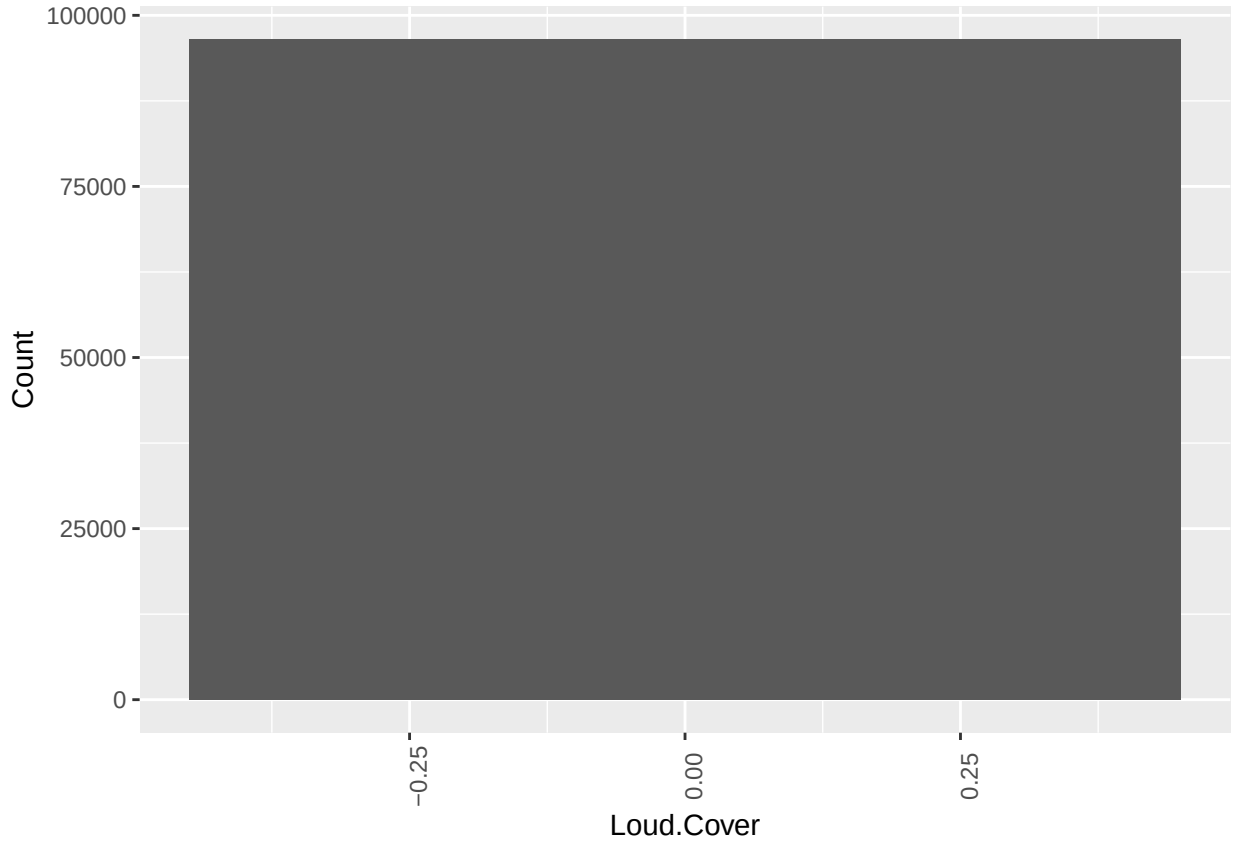

```
weatherHistory %>%
  ggplot(aes(Wind.Bearing..degrees.)) +
  geom_histogram(aes(y = ..density..), fill = "white", color="black") +
  stat_density(kernel = "gaussian", fill = NA, colour = "black")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Quantative discrete

```
weatherHistory %>%
  group_by(Loud.Cover) %>%
  summarize(Count = n()) %>%
  ggplot(aes(x=Loud.Cover, y=Count)) +
  geom_bar(stat='identity', position='dodge') +
  theme(axis.text.x = element_text(angle=90))
```



Task b

First removing all columns that seem irrelevant, reasoning:

- Formatted.Date : When encoded it will be equal to row label (1, 2, 3, . . .) which tells nothing
- Loud.Cover : All values are 0, therfore tells nothing
- Daily.Summary : Too big to onehotencode effectivly

Then remove all rows with NA, do this after removing irrelevant columns so data is not lost to having NA in the removed columns

```
drop_weatherHistory <- weatherHistory %>% dplyr::select(-c("Formatted.Date", "Daily.Summary", "Loud.Cove
drop_weatherHistory <- na.omit(drop_weatherHistory) # Remove all NA
head(drop_weatherHistory)
```

```
##         Summary Precip.Type Temperature..C. Apparent.Temperature..C. Humidity
## 1 Partly Cloudy        rain        9.472222                 7.388889     0.89
## 2 Partly Cloudy        rain        9.355556                 7.227778     0.86
## 3 Mostly Cloudy        rain        9.377778                 9.377778     0.89
## 4 Partly Cloudy        rain        8.288889                 5.944444     0.83
## 5 Mostly Cloudy        rain        8.755556                 6.977778     0.83
## 6 Partly Cloudy        rain        9.222222                 7.111111     0.85
##   Wind.Speed..km.h. Wind.Bearing..degrees. Visibility..km. Pressure..millibars.
## 1           14.1197                    251         15.8263              1015.13
## 2           14.2646                    259         15.8263              1015.63
## 3            3.9284                    204         14.9569              1015.94
## 4           14.1036                    269         15.8263              1016.41
## 5           11.0446                    259         15.8263              1016.51
```

```
## 6           13.9587                     258         14.9569              1016.66
```

```r
num_wH <- drop_weatherHistory %>%
  dplyr::select(-c("Summary", "Precip.Type"))
num_stand_wH <- as.data.frame(sapply(num_wH, function(x) ((x-mean(x))/sd(x))))

qualitative_wH <- drop_weatherHistory %>%
  dplyr::select(c("Summary", "Precip.Type")) #Omitted "Formatted.Date","Daily.Summary"

q1 <- table(1:nrow(drop_weatherHistory), drop_weatherHistory$Precip.Type) # as.data.frame.matrix(
q2 <- table(1:nrow(drop_weatherHistory), drop_weatherHistory$Summary)
q <- as.data.frame.matrix(cbind(q1, q2))

cleaned_wH <- cbind(num_stand_wH, q)
head(cleaned_wH)
```

```
##   Temperature..C. Apparent.Temperature..C.  Humidity Wind.Speed..km.h.
## 1      -0.2575977               -0.3240338 0.7934663        0.47863251
## 2      -0.2698121               -0.3390953 0.6399922        0.49959129
## 3      -0.2674856               -0.1381015 0.7934663       -0.99546821
## 4      -0.3814869               -0.4590684 0.4865181        0.47630376
## 5      -0.3326292               -0.3624667 0.4865181        0.03384067
## 6      -0.2837715               -0.3500020 0.5888342        0.45534498
##   Wind.Bearing..degrees. Visibility..km. Pressure..millibars. null rain snow
## 1              0.5912529        1.306969            0.1016847    0    1    0
## 2              0.6657523        1.306969            0.1059593    0    1    0
## 3              0.1535690        1.099580            0.1086095    0    1    0
## 4              0.7588766        1.306969            0.1126276    0    1    0
## 5              0.6657523        1.306969            0.1134826    0    1    0
## 6              0.6564399        1.099580            0.1147649    0    1    0
##   Breezy Breezy and Dry Breezy and Foggy Breezy and Mostly Cloudy
## 1      0              0               0                        0
## 2      0              0               0                        0
## 3      0              0               0                        0
## 4      0              0               0                        0
## 5      0              0               0                        0
## 6      0              0               0                        0
##   Breezy and Overcast Breezy and Partly Cloudy Clear
## 1                   0                        0     0
## 2                   0                        0     0
## 3                   0                        0     0
## 4                   0                        0     0
## 5                   0                        0     0
## 6                   0                        0     0
##   Dangerously Windy and Partly Cloudy Drizzle Dry Dry and Mostly Cloudy
## 1                                   0       0   0                    0
## 2                                   0       0   0                    0
## 3                                   0       0   0                    0
## 4                                   0       0   0                    0
## 5                                   0       0   0                    0
## 6                                   0       0   0                    0
##   Dry and Partly Cloudy Foggy Humid and Mostly Cloudy Humid and Overcast
## 1                     0     0                      0                  0
## 2                     0     0                      0                  0
## 3                     0     0                      0                  0
```

```
## 4                     0     0                     0                 0
## 5                     0     0                     0                 0
## 6                     0     0                     0                 0
##   Humid and Partly Cloudy Light Rain Mostly Cloudy Overcast Partly Cloudy Rain
## 1                       0          0             0        0              1    0
## 2                       0          0             0        0              1    0
## 3                       0          0             1        0              0    0
## 4                       0          0             0        0              1    0
## 5                       0          0             1        0              0    0
## 6                       0          0             0        0              1    0
##   Windy Windy and Dry Windy and Foggy Windy and Mostly Cloudy
## 1     0             0               0                       0
## 2     0             0               0                       0
## 3     0             0               0                       0
## 4     0             0               0                       0
## 5     0             0               0                       0
## 6     0             0               0                       0
##   Windy and Overcast Windy and Partly Cloudy
## 1                  0                       0
## 2                  0                       0
## 3                  0                       0
## 4                  0                       0
## 5                  0                       0
## 6                  0                       0
```

```
sample <- sample(c(T, F), nrow(cleaned_wH), replace=T, prob=c(0.75, 0.25))
test_wH <- cleaned_wH[!sample,]
train_wH <- cleaned_wH[sample,]
```

Task c

Reason for chosen variables:

- Temperate (C) : Baseline that gets moved
- Humidity : Feels a lot hotter when its more humid, harder to sweat
- Wind speed : Wind makes skin feel colder
- Pressure : Pressure changes based on if it may rain or not, feels different
- Rain/Snow : If it rains the air feels colder

```
wH_lm <- train_wH %>%
  lm(Apparent.Temperature..C. ~ rain + snow + Pressure..millibars. + Humidity + Temperature..C. + Wind.S

summary.aov(wH_lm)
```

```
##                         Df Sum Sq Mean Sq   F value Pr(>F)
## rain                     1  22039   22039 2.188e+06 <2e-16 ***
## snow                     1    839     839 8.326e+04 <2e-16 ***
## Pressure..millibars.     1      1       1 7.161e+01 <2e-16 ***
## Humidity                 1  16944   16944 1.682e+06 <2e-16 ***
## Temperature..C.          1  31066   31066 3.084e+06 <2e-16 ***
## Wind.Speed..km.h.        1    252     252 2.501e+04 <2e-16 ***
## Residuals            72273    728       0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(wH_lm)
```

```
##
## Call:
## lm(formula = Apparent.Temperature..C. ~ rain + snow + Pressure..millibars. +
##     Humidity + Temperature..C. + Wind.Speed..km.h., data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.39578 -0.06856 -0.00933  0.06215  0.50045
##
## Coefficients:
##                         Estimate Std. Error  t value Pr(>|t|)
## (Intercept)            0.0427527  0.0050448    8.475  < 2e-16 ***
## rain                  -0.0382702  0.0050625   -7.560 4.09e-14 ***
## snow                  -0.0773794  0.0052193  -14.826  < 2e-16 ***
## Pressure..millibars.   0.0020301  0.0003722    5.455 4.92e-08 ***
## Humidity               0.0161352  0.0005155   31.300  < 2e-16 ***
## Temperature..C.        0.9962280  0.0005943 1676.248  < 2e-16 ***
## Wind.Speed..km.h.     -0.0622154  0.0003934 -158.141  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1004 on 72273 degrees of freedom
## Multiple R-squared:  0.9899, Adjusted R-squared:  0.9899
## F-statistic: 1.177e+06 on 6 and 72273 DF,  p-value: < 2.2e-16
```

As can be seen an the ANOVA and t test for the different values, they are all significant within $\alpha \approx 0$ which means that there is almost 0 chance that the factors are due to random chance. (FIX LATER, DOUBBLE CHECK)

```r
y_test_true <- test_wH$Apparent.Temperature..C.
y_test_pred <- predict(wH_lm, newdata = test_wH)
y_train_true <- train_wH$Apparent.Temperature..C.
y_train_pred <- predict(wH_lm, newdata = train_wH)
```

RMSE

$$\text{RMSE}(y, \hat{y}) = \sqrt{\frac{\sum_{i=0}^{N-1}(y_i - \hat{y}_i)^2}{N}}$$

```r
rmse_test <- RMSE(y_test_pred, y_test_true)
rmse_train <- RMSE(y_train_pred, y_train_true)
rmse_test
```

```
## [1] 0.1003436
```

```r
rmse_test
```

```
## [1] 0.1003436
```

MAE

$$\text{MAE}(x, y) = \sum_{i=1}^{D} |x_i - y_i|$$

```
mae_test <- MAE(y_test_pred, y_test_true)
mae_train <- MAE(y_train_pred, y_train_true)
mae_test
```

```
## [1] 0.07892357
```

```
mae_train
```

```
## [1] 0.07901741
```

R^2 score (coefficient of determination)

$$R^2 = 1 - \frac{\text{SSR (sum of square regression)}}{\text{SST (total sum of squares)}} = 1 - \frac{\sum_{i=0}^{N-1}(y_i - \hat{y}_i)^2}{\sum_{i=0}^{N-1}(y_i - \hat{y})^2}$$

```
R2_test <- R2(y_test_pred, y_test_true)
R2_train <- R2(y_train_pred, y_train_true)
R2_test
```

```
## [1] 0.9901021
```

```
R2_train
```

```
## [1] 0.9898688
```

Exercise 3 - Linear Regression and Diagnostic Plots

Task a

a) Linearity

The relationship between X and Y is linear.

If a function is linear, then all factors increase at a consistent rate for each step. Gradient and all partial first order derivatives will be a function of constants Linear function: $y(x, z) = 2x + 4z + 5xz + 10$ Non-Linear function: $y(x, z) = 2x^2 + 5z^5$

b) Homoscedasticity

Residuals have constant variance.

homogeneity of variance assumes that all observations are picked from a sources that have equal variance. In other words the data points around a linear model should vary equally from the line, if there are any cone shape or other such irregularities this assumption is broken and the model that will be produced will eb flawed.

c) Independence

Residuals are independent.

Observations can't depend on each other, in other words if you pick one observation from a population for your sample, this should not affect the next sample you choose. In other words no observations should depend or affect each other.

By the very nature of lm, it is **assumed** that you have a i.i.d dataset. This has to be done in the sampeling stage, not the cleaning stage.

d) Normaility

Residuals are normally distributed.

Assumes that the data follows a normal distribution.

Can be tested by making a Q-Q plot and seeing how well the data points follows the line. If the right tail is very heavy, you probably should log the value slightly heavy tails can still be used because of the law of large numbers.

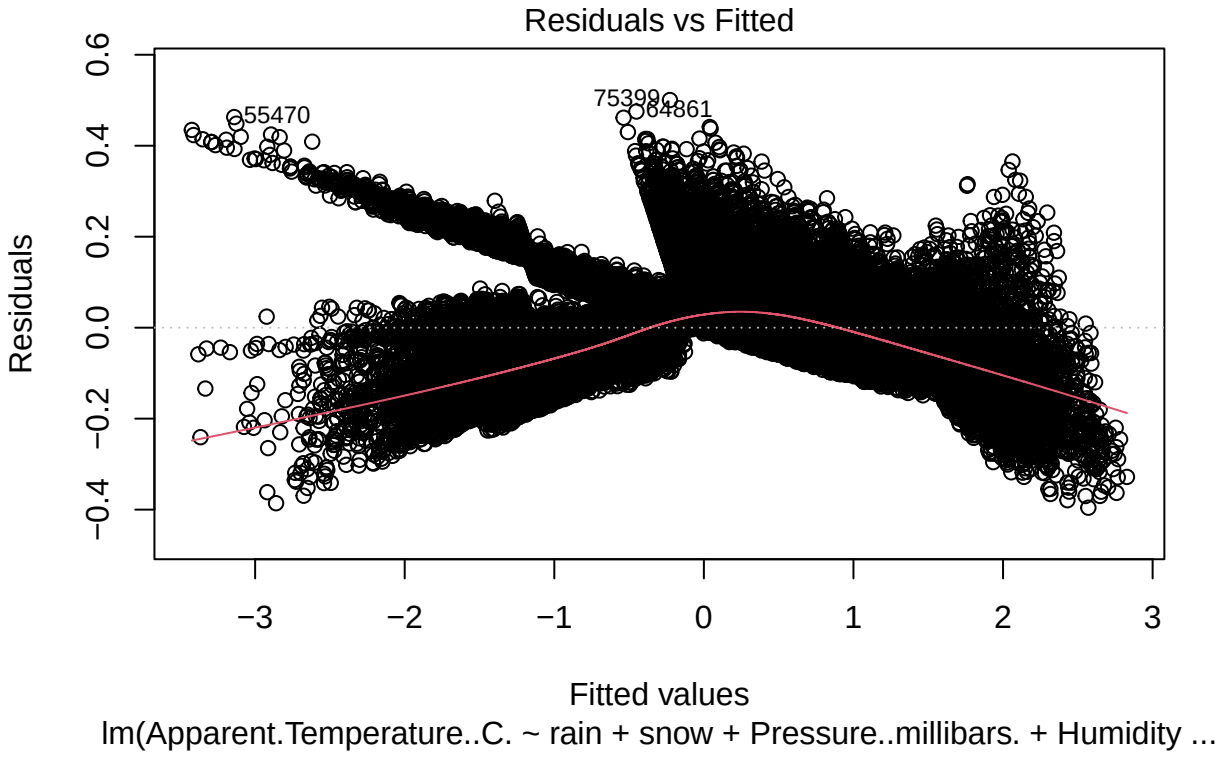
Task b

These are all diagnostic plots that allow us to test our assumptions

Residuals vs Fitted

A scatter plot where residuals are on the y-axis and fitted values are on the x-axis. Used to detect non-linearity, unequal error variances, and outliers.

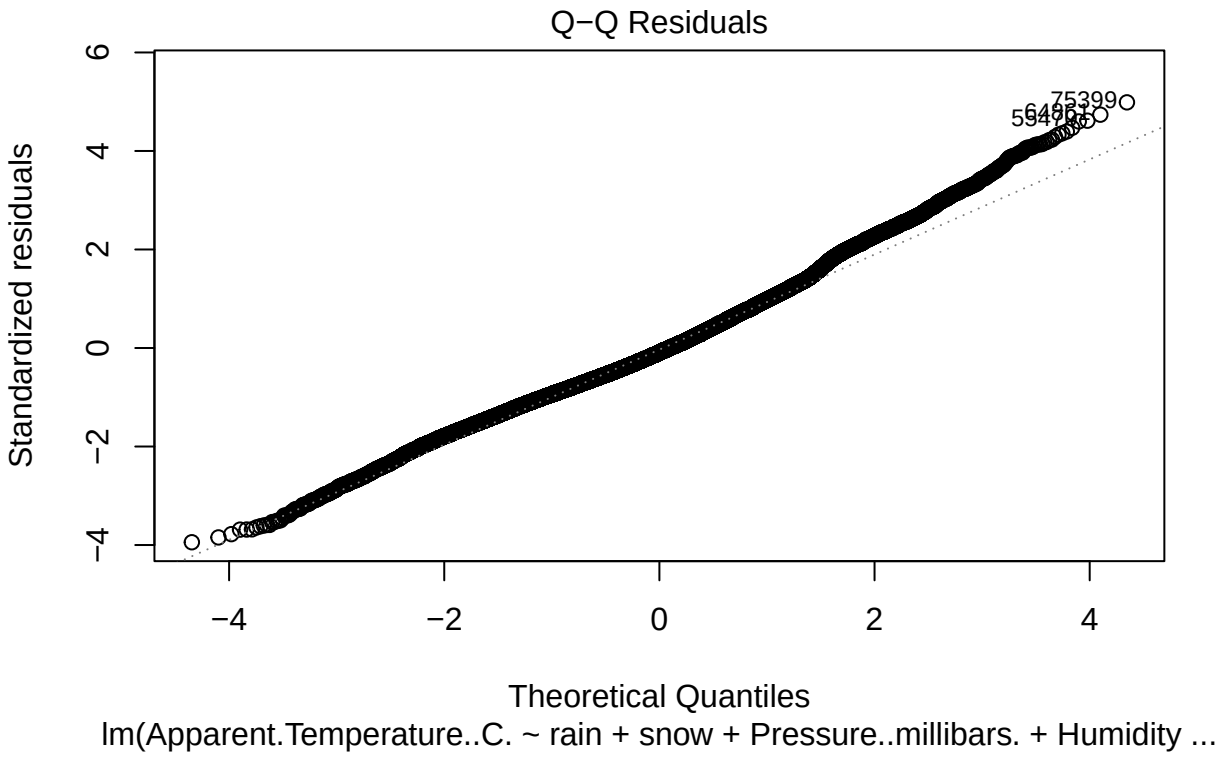
```
plot(wH_lm,1)
```



Normal Q-Q

Points on the Normal Q-Q plot provide an indication of univariate normality of the dataset. If the data is normally distributed, the points will fall on the line, otherwise it implies that the assumption of Normality is broken. To test for normality of residuals

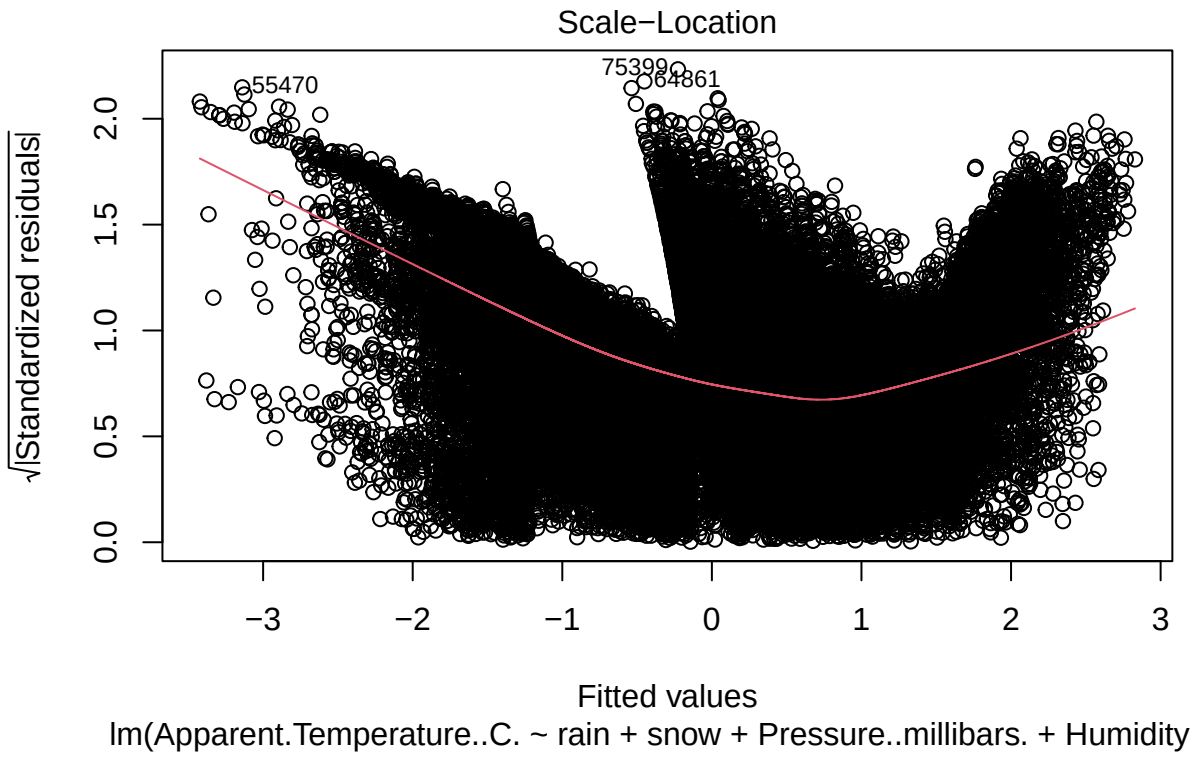
```
plot(wH_lm,2)
```



Scale-Location (or Spread-Location)

Simmilar to residuals vs fitted, but instead of using residuals on the y-axis it uses the square root of the residuals. Used to check for the assumption of homoscedascity. If the line is roughly horizontal and there is no clear pattern (like a cone) in the scatter plot then homoscedacity is lokely satisfied.

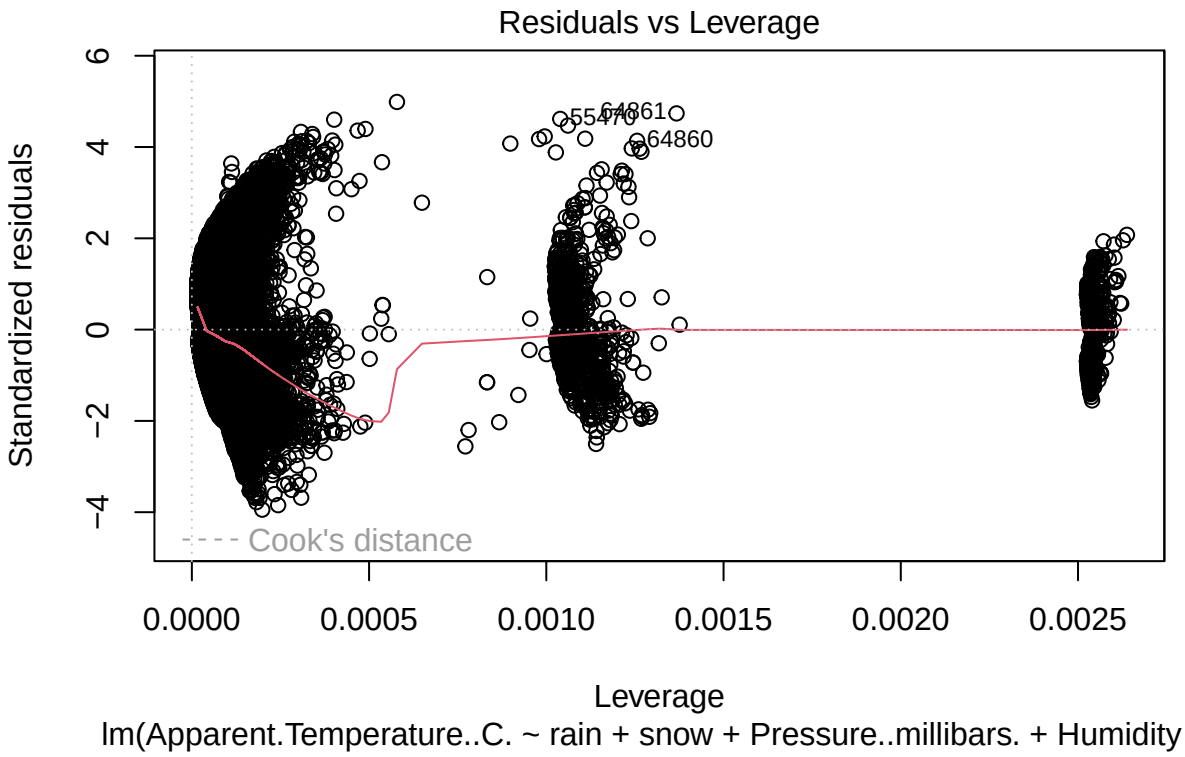
```
plot(wH_lm,3)
```



Residuals vs Leverage

Allows us to identify influential observations **Leverage:** extent to which the coefficients in the regression model would change if a particular observations was removed from the dataset (outliers). **Standarized residuals:** standardized difference between a predicted value for an observation and the actual value of the observation.

```
plot(wH_lm,5)
```



Task c

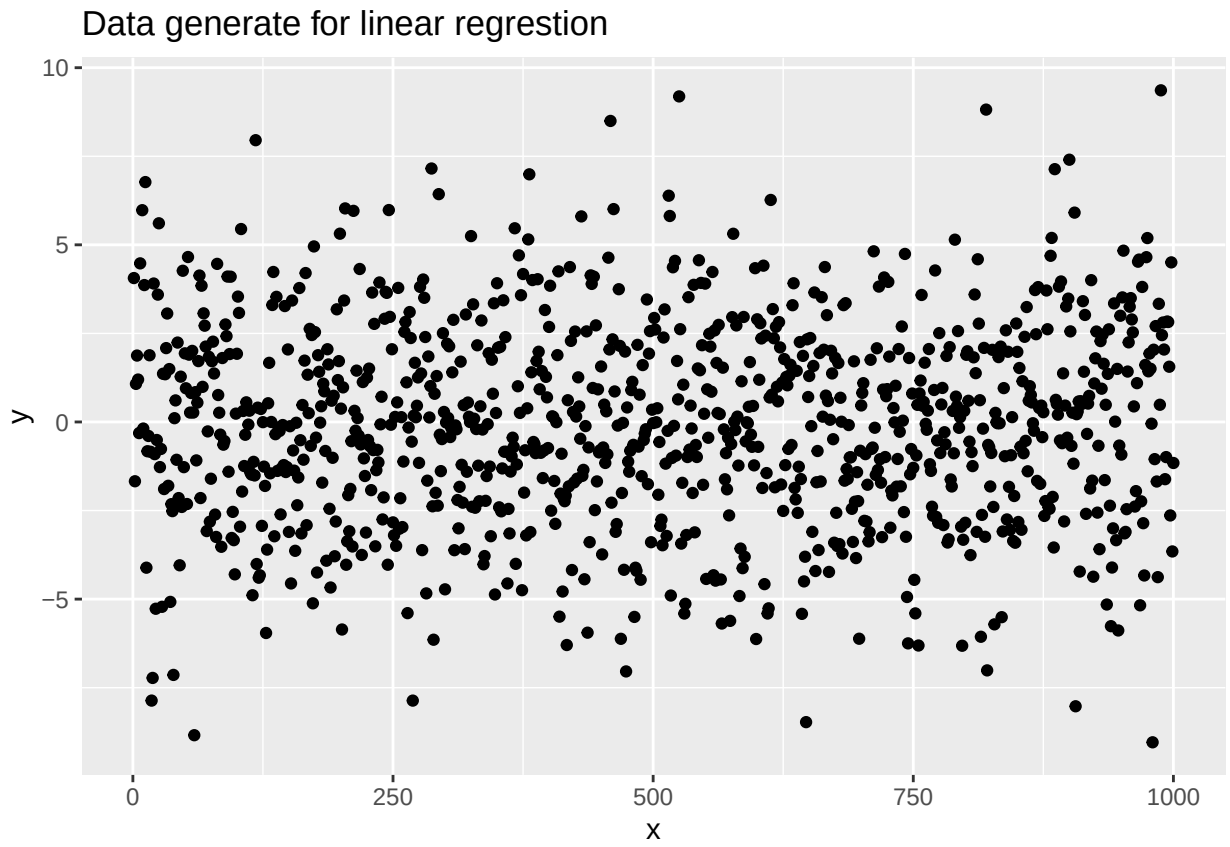
```
data_gen <- function(c=0, t=0, cm=0, cp=100, cs=0, nnm=1, nnp=10000, snn=500, nnnm=0, nnnp=10000, snnn=5
  set.seed(42)

  n <- 1000
  x <- 1:n

  # Changeable parameters
  # - Change the parameters to affect the generated data points below.
  # - You may copy this code multiple times to answer all the questions in the exercise.
  # - You may find it reasonable to argue for multiple violations from a single generated set of data p

  contant <- c
  trend <- t
  curve_magnitue <- cm
  curve_period <- cp
  curve_shift <- cs
  normal_noise_magnitue <- nnm
  norm_noise_periode <- nnp
  shift_norm_noise <- snn
  non_normal_noise_magnitue <- nnnm
  non_norm_noice_periode <- nnnp
  shift_non_norm_noise <- snnn
```

```
  y.gen <- contant +
    trend * x +
    curve_magnitue* sin(
      (x/curve_period + curve_shift)*pi
      ) +
    normal_noise_magnitue*cos(
      (x/norm_noise_periode + shift_norm_noise/norm_noise_periode)*pi
      )*rnorm(n, sd = 3) +
    non_normal_noise_magnitue*cos(
      (x/non_norm_noice_periode + shift_non_norm_noise/non_norm_noice_periode)*pi
      ) * rexp(n, rate = 0.2)

  p <- qplot(x, y.gen, ylab = "y") +
    geom_point(size = 0.1) +
    labs(title = "Data generate for linear regrestion")

  # Display the plot
  print(p)
  return(list("x"=x,"y.gen"=y.gen))
}
```

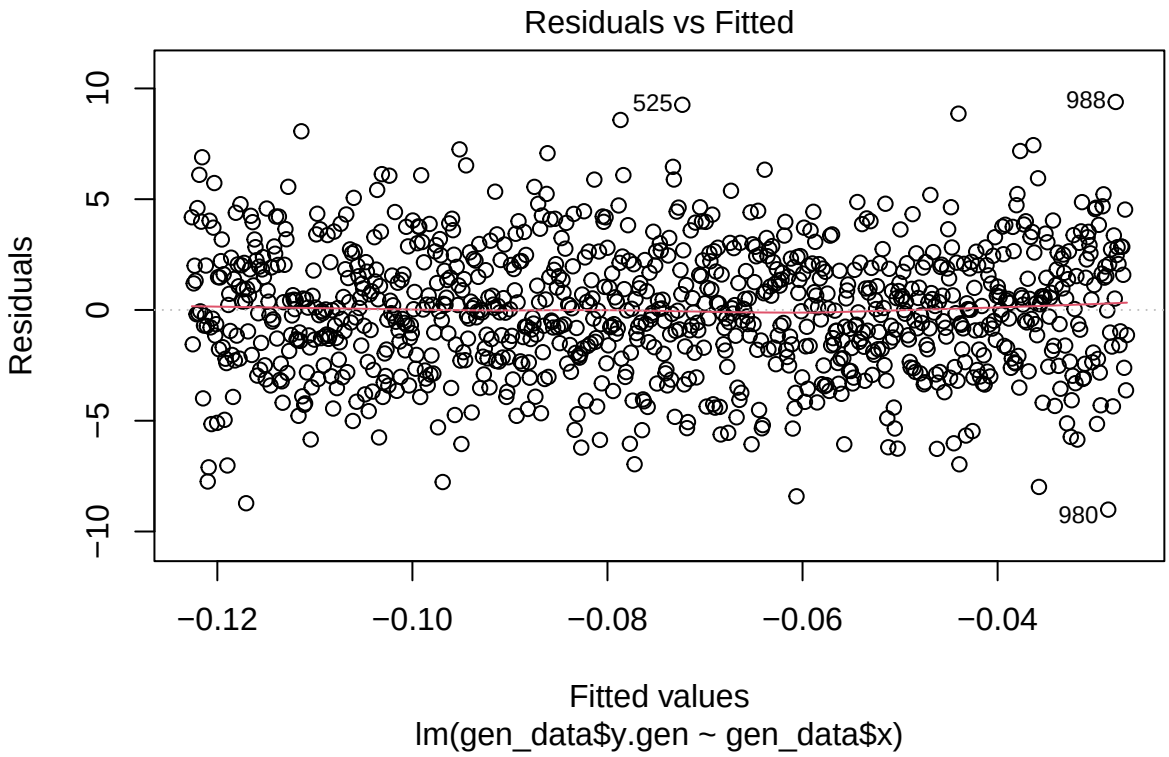
Holds all the assumptions

```
gen_data <- data_gen()
```

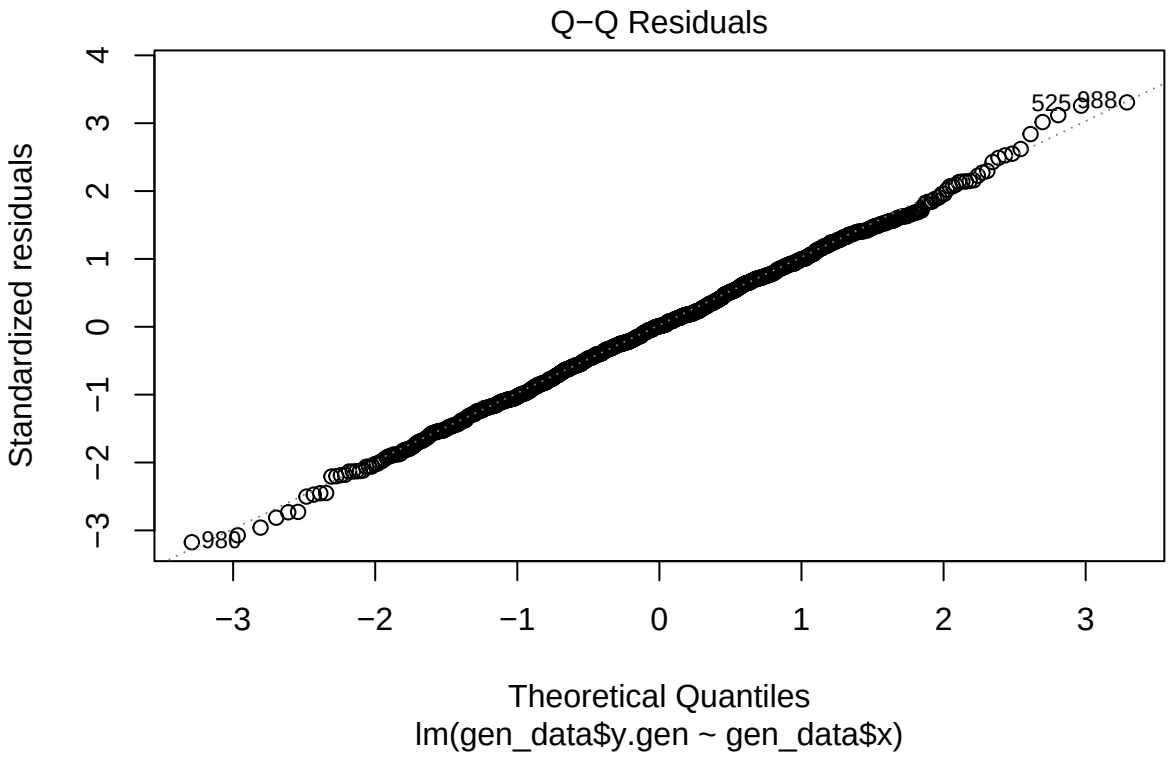
```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



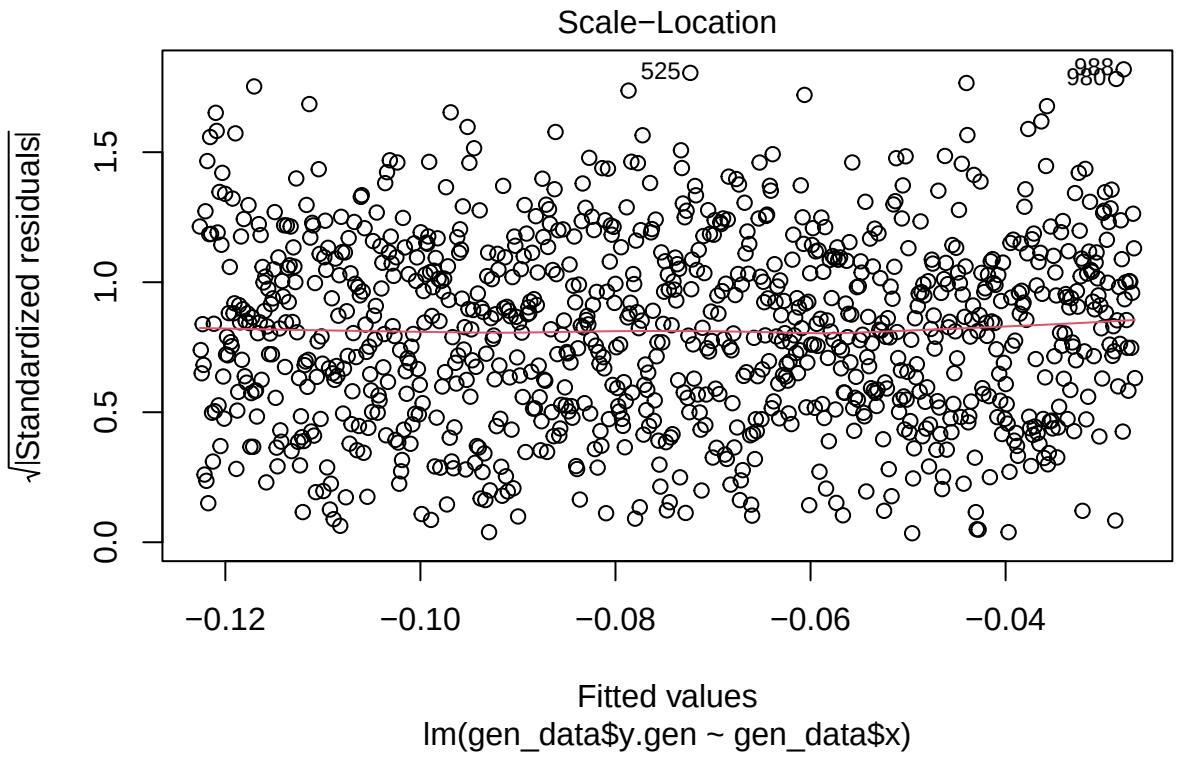

```
lm.gen <- lm(gen_data$y.gen ~ gen_data$x)
plot(lm.gen, which = 1)
```



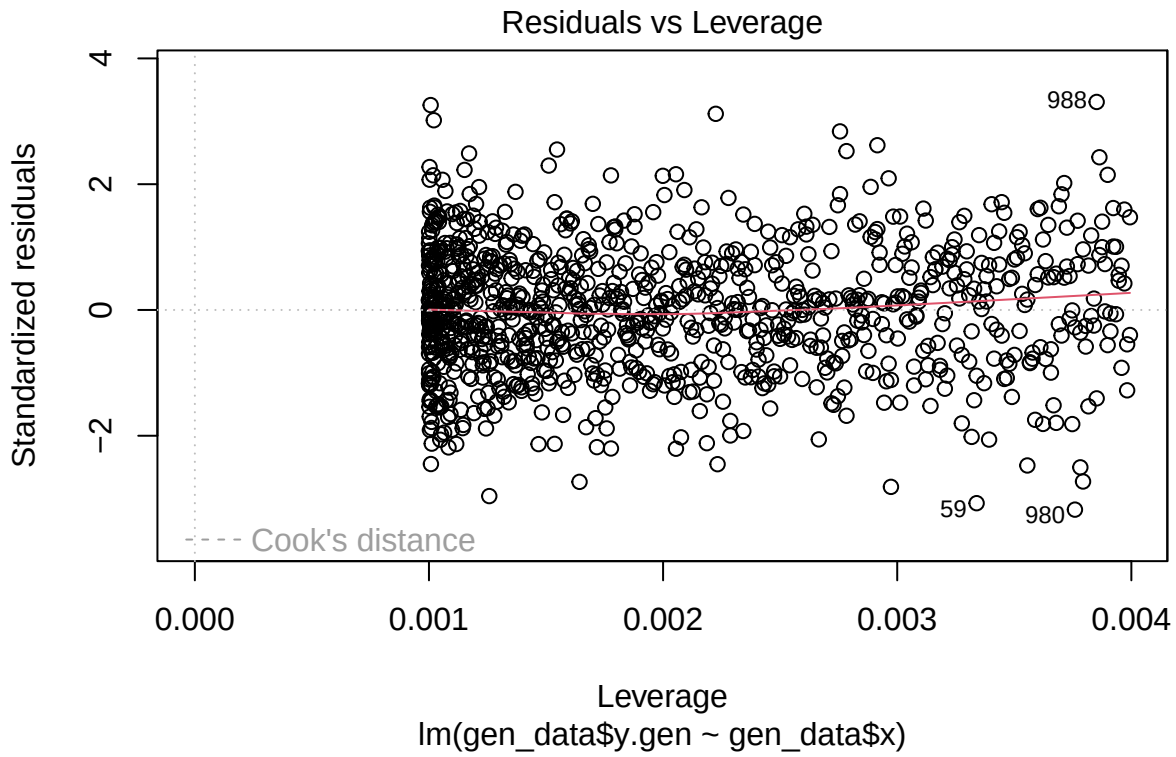

```
plot(lm.gen, which = 2)
```




```
plot(lm.gen, which = 3)
```

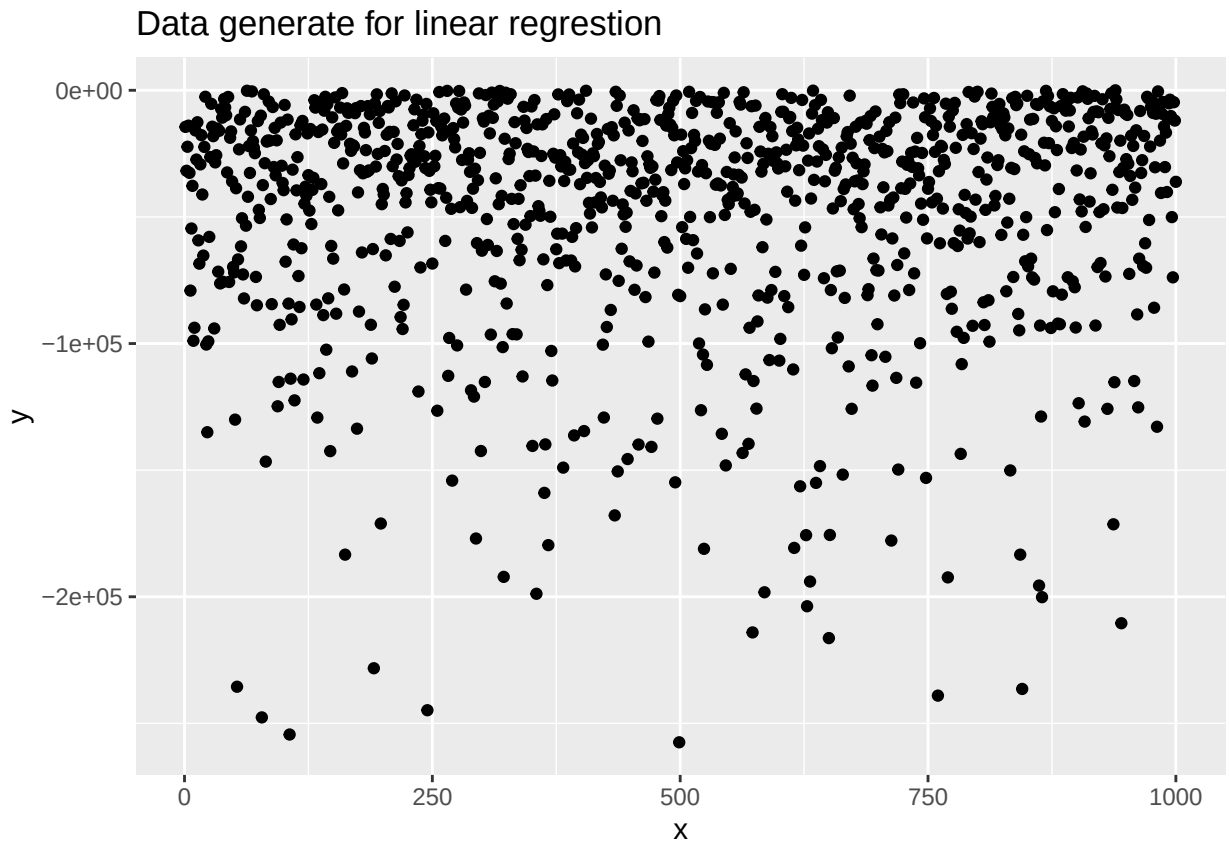



```
plot(lm.gen, which = 5)
```

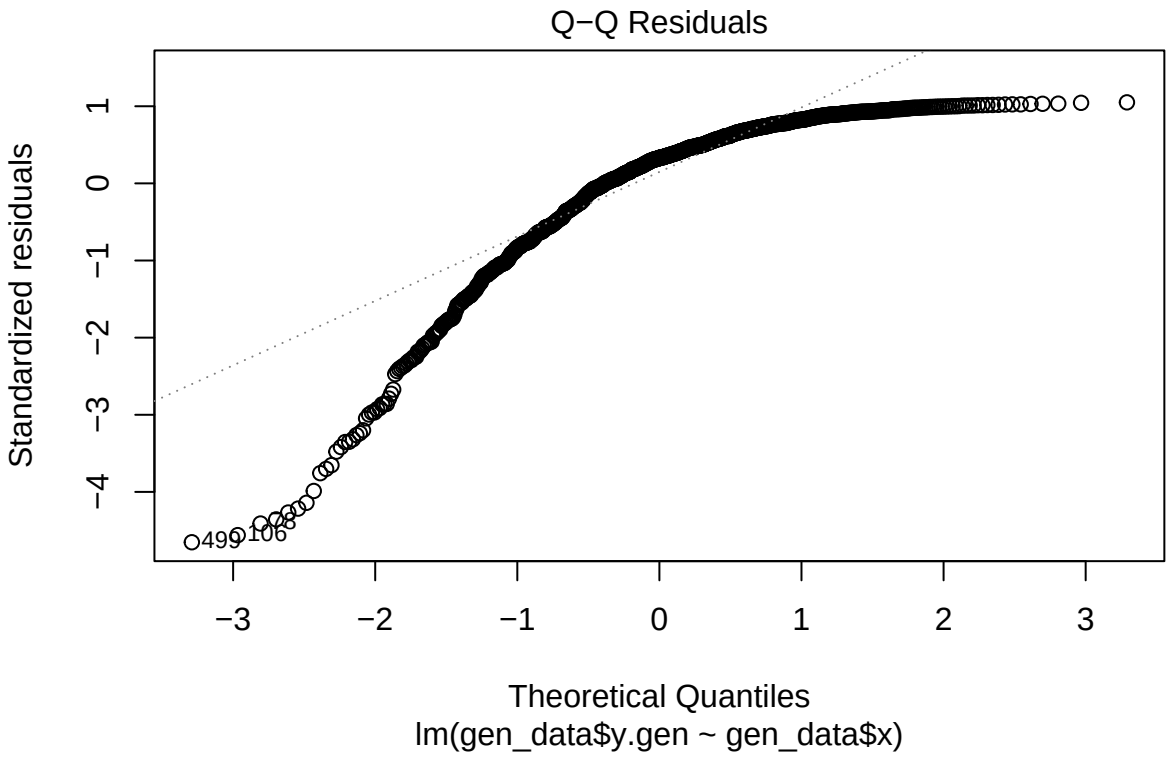


Breaks the assumption of Homoscedastiicty

```
gen_data <- data_gen(nnnm = -10000, snnn = 5000, nnnp = 50000)
```

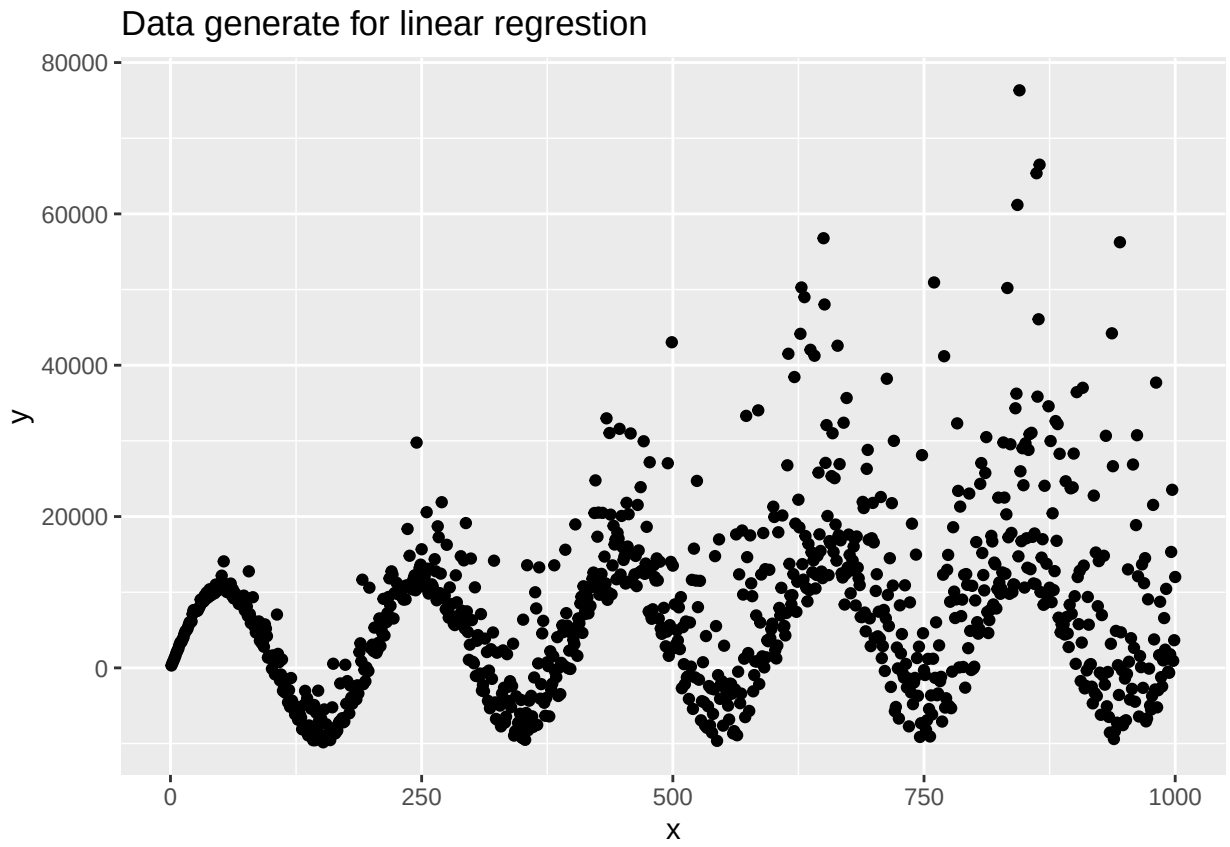



```
lm.gen <- lm(gen_data$y.gen ~ gen_data$x)
plot(lm.gen, which = 2)
```

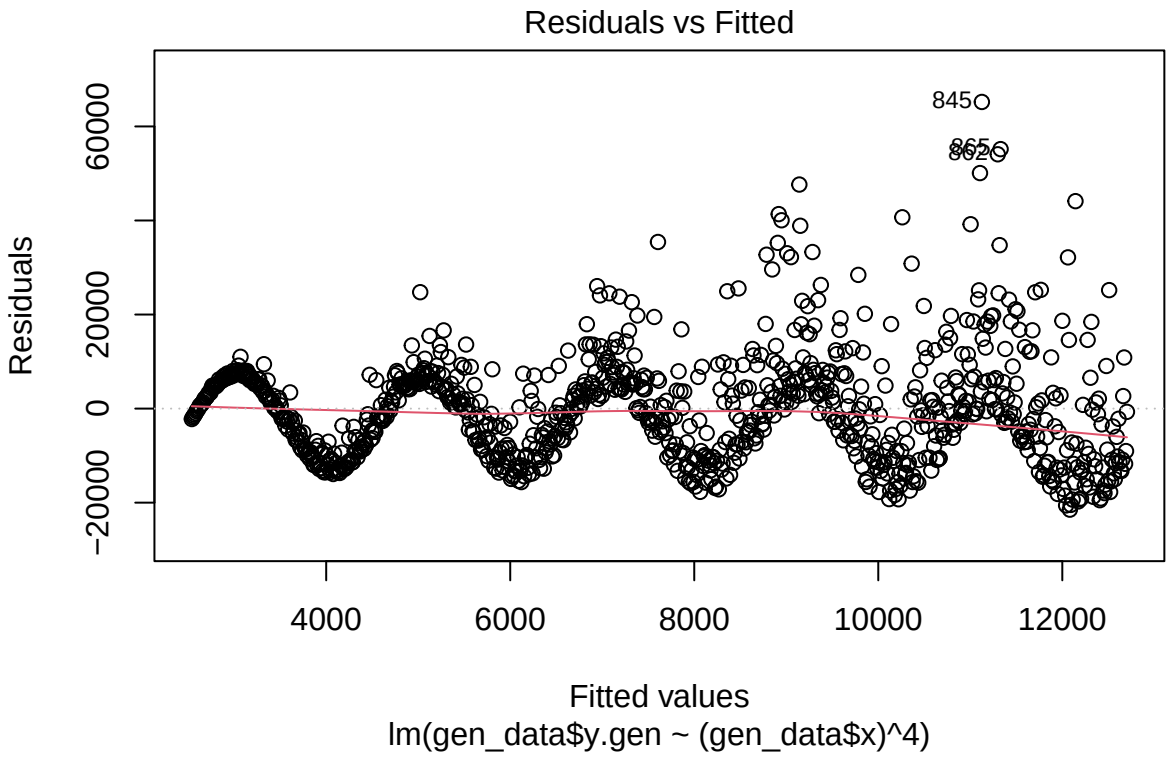


Breaks the assumption of Linearity

```
gen_data <- data_gen(nnnm = -10000, snnn = 5000, cs=10000, cm = 10000, snn = 10000)
```

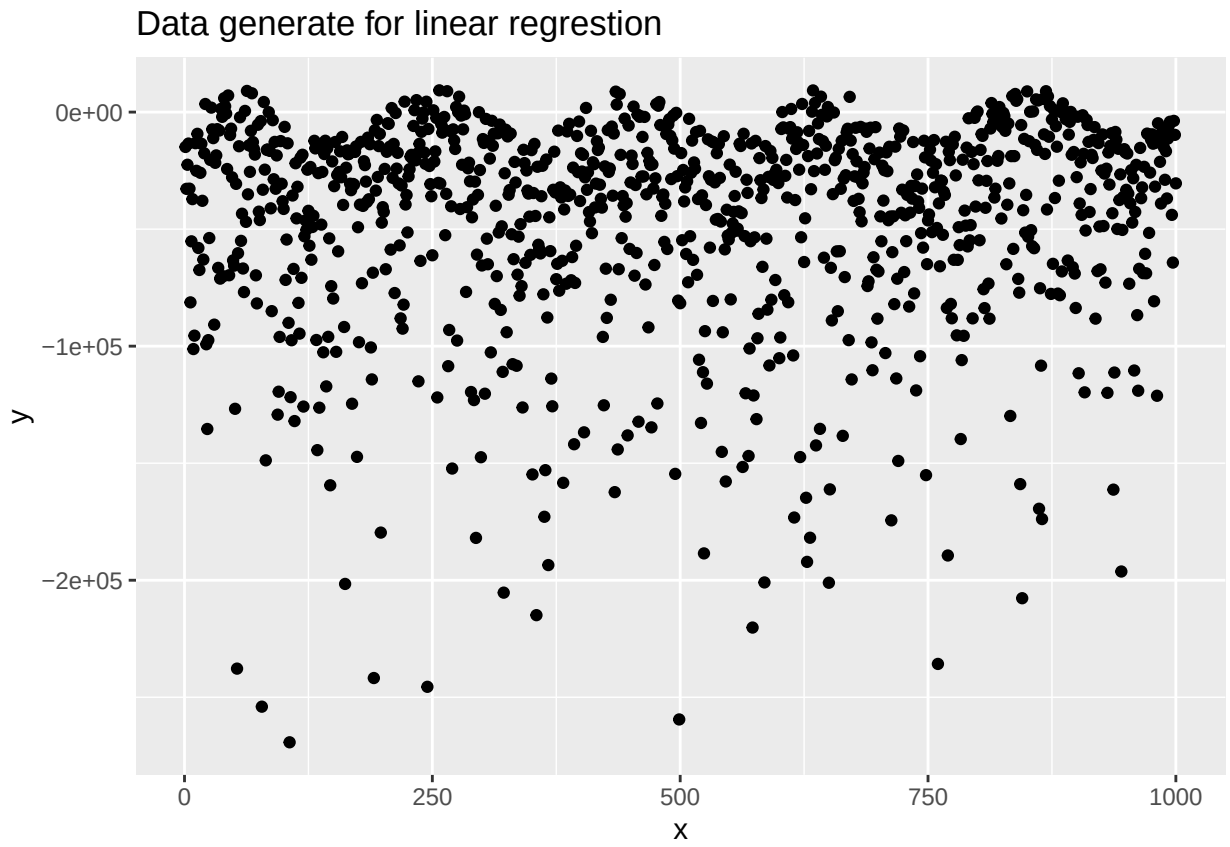



```
lm.gen <- lm(gen_data$y.gen ~ (gen_data$x)^4)
plot(lm.gen, 1)
```

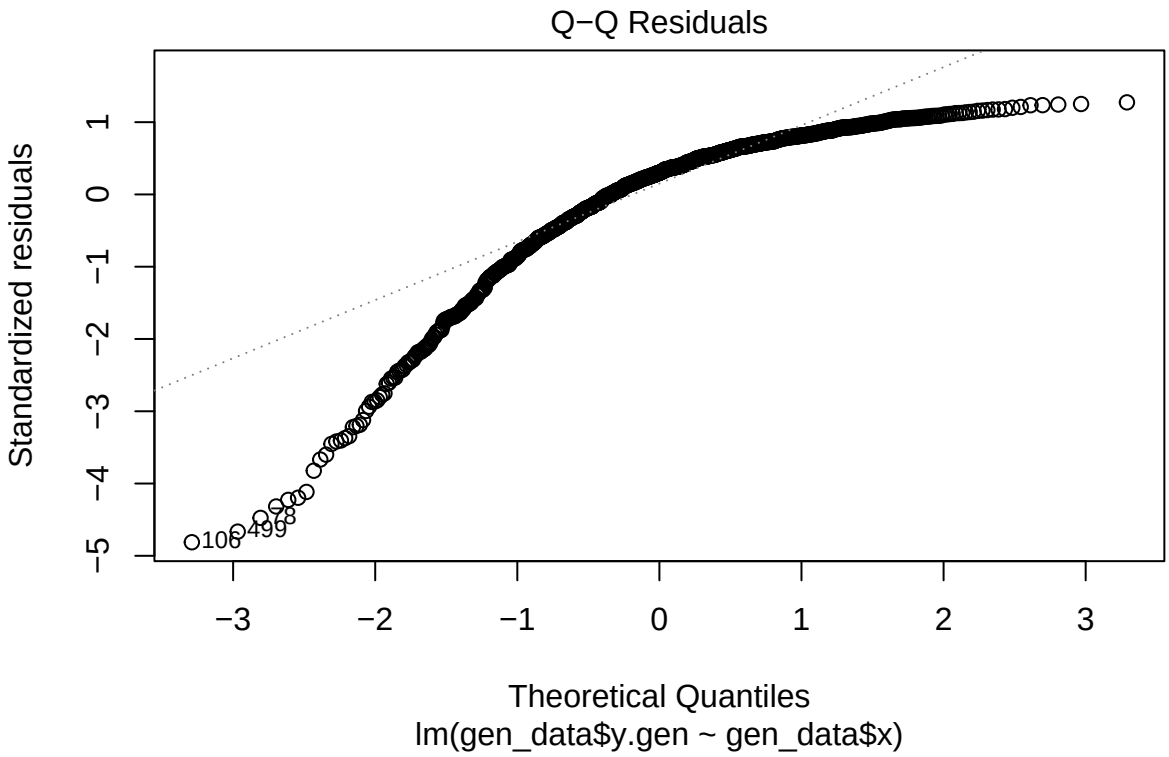


Breaks the assumption of Normality.

```
gen_data <- data_gen(nnnm = 10000, snnn = 5000, cs=10000, cm = 10000, snn = 10000, nnm = 0, nnnp = 5000
```




```
lm.gen <- lm(gen_data$y.gen ~ gen_data$x)
plot(lm.gen, 2)
```



Task d

Breaks Homoscedasticity

The error distibution is not consistent, aka

$$\epsilon_i \neq \sigma^2$$

instead it is

$$\sigma_i^2 = x_i \sigma^2$$

, in other words heteroscedastic.

If plot X against Y you will see a cone shape instead of expected line, clearly there are different distributions of variances based on X

Breaks Linearity

The Residual vs Fitted shows a clear cone as well as a clear non-linear pattern between X and Y

Breaks Normality

The Q-Q Normal plot clearly shows high bias towards the tails

Exercise 4 - correlation and partial correlation

Task a

Correlation: Degree to which a pair of variables/factors are linearly related. As in the increase in one variable either increases or decreases another. Correlation does not imply causation, even if Smoking and

higher life expectancy is higher in one area, that doesn't automatically imply that smoking is healthy.

Higher variance decreases correlation, while higher covariance increases it. In other words the more the two variables/factors trend and the less variance from that trend they have the more correlated the two varaibles/factors are.

Task b

Partial correlation: measures the degree of association between two random variables, with the effect of a set of controlling random variables removed. Wiki Helps to understand the relationship between two variables, controlling for the effect of one or more additional variables.

The formula describes the correlation between the residuals e_X and e_Y resulting from the linear regression of X with Z and of Y with Z, respectively.

Scenarios

1. You suspect there are inter dependencies between X=Study hours per week, Y=Exam score and Z=IQ
2. You suspect there are inter dependencies between X=Calories consumed, Y=Weight loss and Z=Hours of intense exercise.
3. You suspect there are inter dependencies between X=Time spent on social media, Y=Self-esteem and Z=Age

Task c

Yes, the property of correlation regarding scaling holds for partial correlation as well. Just like regular correlation, partial correlation is not affected by scaling of the variables, only the direction or sign can change. Let's do the math so see how this is the case:

1. In the partial correlation formula we have: Pxy Pxz Pyz
2. if we then scale the x and y variables with a1 and a2 we get: Pxy = sgn(a1,a2)*Pxy Pxz = sgn(a1)*Pxz Pyz = sgn(a2)*Pyz
3. Then we substitute the coefficients with the scaled ones and obtain the formula: Pxy|z = (Pxy - Pxz*Pyz) / (sqrt(1-(Pxz)^2)*sqrt(1-(Pyz)^2) -> (sgn(a1,a2)*Pxy - sgn(a1)*Pxz*sgn(a2)*Pyz) / (sqrt(1-(sgn(a1)*Pxz)^2)*sqrt(1-(sgn(a2)*Pyz)^2)
4. after doing the math and simplifying the formula we obtain sgn(a1,a2)*Pxy|z

Task d

```
weatherHistory <- read.csv("weatherHistory.csv")
select_wH <- weatherHistory %>% dplyr::select(Temperature..C., Apparent.Temperature..C., Humidity)
```

```
pairwise_corr <- select_wH %>%
  cor() %>%
  data.frame()
pairwise_corr
```

```
##                          Temperature..C. Apparent.Temperature..C.   Humidity
## Temperature..C.                1.0000000                0.9926286 -0.6322547
## Apparent.Temperature..C.       0.9926286                1.0000000 -0.6025710
## Humidity                      -0.6322547               -0.6025710  1.0000000
```

```
part_corr <- select_wH %>%
  pcor()
```

```
part_corr
```

```
## $estimate
##                           Temperature..C. Apparent.Temperature..C.   Humidity
## Temperature..C.                 1.0000000                0.9892298 -0.3528185
## Apparent.Temperature..C.        0.9892298                1.0000000  0.2664916
## Humidity                       -0.3528185                0.2664916  1.0000000
##
## $p.value
##                           Temperature..C. Apparent.Temperature..C. Humidity
## Temperature..C.                         0                        0        0
## Apparent.Temperature..C.                0                        0        0
## Humidity                                0                        0        0
##
## $statistic
##                           Temperature..C. Apparent.Temperature..C.   Humidity
## Temperature..C.                    0.0000               2098.90993 -117.10342
## Apparent.Temperature..C.        2098.9099                  0.00000   85.86793
## Humidity                        -117.1034                 85.86793    0.00000
##
## $n
## [1] 96453
##
## $gp
## [1] 1
##
## $method
## [1] "pearson"
```

- Apparent.Temperature..C. and Temperature..C. has a partial correlation value of ca 0.99 which implies that they are highly consistent with each other and increase with each other. Very low difference between pairwise to partial where they are equal if rounded to nearest 2 decimal, which implies that Z=Humidity has very little effect on them.
- Humidity and Temperature..C. has a low partial correlation value of -0.35 which implies that they are not consistent with each other and decreases when the other increases. Decreases from -0.63 to -0.35 which implies that Z=Apparent.Temperature.C.. is affecting them. Z is a significant confounder.
- Humidity and Apparent.Temperature..C. has a low partial correlation value of 0.27 which implies that they are not consistent with each other and increase with each other. Dropped from pairwise correlation -0.60 to 0.27 which implies that Z=Temperature..C. is significantly affecting them. Z is a significant cofunder.