EDA of Diabetes130US Dataset UCI

Shivansh Sharma
ss14890
shivansh.sharma@nyu.edu
Fall, 2021

Table of Contents

Getting to know the Dataset

Data Source: https://archive.ics.uci.edu/ml/datasets/diabetes+130-us+hospitals+for+years+1999-2008

Number of Instances	101766
Number of Attributes	49
Number of Classes	3

Class Attribute Name: **Readmitted** ("Days to inpatient readmission.")

"<30"	if the patient was readmitted in less than 30 days
">30"	if the patient was readmitted in more than 30 days
"No"	for no record of readmission.

Features List with Description:

Seq no.	Feature name	Type	Description and values
1	Encounter ID	Numeric	Unique identifier of an encounter
2	Patient number	Numeric	Unique identifier of a patient
3	Race	Nominal	Values: Caucasian, Asian, African American, Hispanic, and other
4	Gender	Nominal	Values: male, female, and unknown/invalid
5	Age	Nominal	Grouped in 10-year intervals: 0, 10), 10, 20), …, 90, 100)
6	Weight	Numeric	Weight in pounds.
7	Admission type	Nominal	Integer identifier corresponding to 9 distinct values, for example, emergency, urgent, elective, newborn, and not available
8	Discharge disposition	Nominal	Integer identifier corresponding to 29 distinct values, for example, discharged to home, expired, and not available
9	Admission source	Nominal	Integer identifier corresponding to 21 distinct values, for example, physician referral, emergency room, and transfer from a hospital
10	Time in hospital	Numeric	Integer number of days between admission and discharge
11	Payer code	Nominal	Integer identifier corresponding to 23 distinct values, for example, Blue Cross/Blue Shield, Medicare, and self-pay
12	Medical specialty	Nominal	Integer identifier of a specialty of the admitting physician, corresponding to 84 distinct values, for example, cardiology, internal medicine, family/general practice, and surgeon
13	Number of lab procedures	Numeric	Number of lab tests performed during the encounter
14	Number of procedures	Numeric	Number of procedures (other than lab tests) performed during the encounter

15	Number of medications	Numeric	Number of distinct generic names administered during the encounter
16	Number of outpatient visits	Numeric	Number of outpatient visits of the patient in the year preceding the encounter
17	Number of emergency visits	Numeric	Number of emergency visits of the patient in the year preceding the encounter
18	Number of inpatient visits	Numeric	Number of inpatient visits of the patient in the year preceding the encounter
19	Diagnosis 1	Nominal	The primary diagnosis (coded as first three digits of ICD9); 848 distinct values
20	Diagnosis 2	Nominal	Secondary diagnosis (coded as first three digits of ICD9); 923 distinct values
21	Diagnosis 3	Nominal	Additional secondary diagnosis (coded as first three digits of ICD9); 954 distinct values
22	Number of diagnoses	Numeric	Number of diagnoses entered to the system
23	Glucose serum test result	Nominal	Indicates the range of the result or if the test was not taken. Values: “>200,” “>300,” “normal,” and “none” if not measured
24	A1c test result	Nominal	Indicates the range of the result or if the test was not taken. Values: “>8” if the result was greater than 8%, “>7” if the result was greater than 7% but less than 8%, “normal” if the result was less than 7%, and “none” if not measured.
25-47	23 features for medications	Nominal	For the generic names: metformin, repaglinide, nateglinide, chlorpropamide, glimepiride, acetohexamide, glipizide, glyburide, tolbutamide, pioglitazone, rosiglitazone, acarbose, miglitol, troglitazone, tolazamide, examide, sitagliptin, insulin, glyburide-metformin, glipizide-metformin, glimepiride-pioglitazone, metformin-rosiglitazone, and metformin-pioglitazone, the feature indicates whether the drug was prescribed or there was a change in the dosage. Values: “up” if the dosage was increased during the encounter, “down” if the dosage was decreased, “steady” if the dosage did not change, and “no” if the drug was not prescribed
48	Change of medications	Nominal	Indicates if there was a change in diabetic medications (either dosage or generic name). Values: “change” and “no change”
49	Diabetes medications	Nominal	Indicates if there was any diabetic medication prescribed. Values: “yes” and “no”
50	Readmitted	Nominal	Days to inpatient readmission. Values: “<30” if the patient was readmitted in less than 30 days, “>30” if the patient was readmitted in more than 30 days, and “No” for no record of readmission.

Load Data

```
diabetes = read.csv("diabetic_data.csv", stringsAsFactors=F)

dim(diabetes)

## [1] 101766     50

str(diabetes)

##'data.frame':       101766 obs. of  50 variables:
## $ encounter_id            : int  2278392 149190 64410 500364 16680 35754 55842 63768 12522 15738 ...
## $ patient_nbr             : int  8222157 55629189 86047875 82442376 42519267 82637451 84259809 ...
## $ race                    : chr  "Caucasian" "Caucasian" "AfricanAmerican" "Caucasian" ...
## $ gender                  : chr  "Female" "Female" "Female" "Male" ...
## $ age                     : chr  "[0-10)" "[10-20)" "[20-30)" "[30-40)" ...
## $ weight                  : chr  "?" "?" "?" "?" ...
## $ admission_type_id       : int  6 1 1 1 1 2 3 1 2 3 ...
## $ discharge_disposition_id: int  25 1 1 1 1 1 1 1 1 3 ...
## $ admission_source_id     : int  1 7 7 7 7 2 2 7 4 4 ...
## $ time_in_hospital        : int  1 3 2 2 1 3 4 5 13 12 ...
## $ payer_code              : chr  "?" "?" "?" "?" ...
## $ medical_specialty       : chr  "Pediatrics-Endocrinology" "?" "?" "?" ...
## $ num_lab_procedures      : int  41 59 11 44 51 31 70 73 68 33 ...
## $ num_procedures          : int  0 0 5 1 0 6 1 0 2 3 ...
## $ num_medications         : int  1 18 13 16 8 16 21 12 28 18 ...
## $ number_outpatient       : int  0 0 2 0 0 0 0 0 0 0 ...
## $ number_emergency        : int  0 0 0 0 0 0 0 0 0 0 ...
## $ number_inpatient        : int  0 0 1 0 0 0 0 0 0 0 ...
## $ diag_1                  : chr  "250.83" "276" "648" "8" ...
## $ diag_2                  : chr  "?" "250.01" "250" "250.43" ...
## $ diag_3                  : chr  "?" "255" "V27" "403" ...
## $ number_diagnoses        : int  1 9 6 7 5 9 7 8 8 8 ...
## $ max_glu_serum           : chr  "None" "None" "None" "None" ...
## $ A1Cresult               : chr  "None" "None" "None" "None" ...
## $ metformin               : chr  "No" "No" "No" "No" ...
## $ repaglinide             : chr  "No" "No" "No" "No" ...
## $ nateglinide             : chr  "No" "No" "No" "No" ...
## $ chlorpropamide          : chr  "No" "No" "No" "No" ...
## $ glimepiride             : chr  "No" "No" "No" "No" ...
## $ acetohexamide           : chr  "No" "No" "No" "No" ...
## $ glipizide               : chr  "No" "No" "Steady" "No" ...
## $ glyburide               : chr  "No" "No" "No" "No" ...
## $ tolbutamide             : chr  "No" "No" "No" "No" ...
## $ pioglitazone            : chr  "No" "No" "No" "No" ...
## $ rosiglitazone           : chr  "No" "No" "No" "No" ...
## $ acarbose                : chr  "No" "No" "No" "No" ...
## $ miglitol                : chr  "No" "No" "No" "No" ...
## $ troglitazone            : chr  "No" "No" "No" "No" ...
## $ tolazamide              : chr  "No" "No" "No" "No" ...
## $ examide                 : chr  "No" "No" "No" "No" ...
## $ citoglipton             : chr  "No" "No" "No" "No" ...
## $ insulin                 : chr  "No" "Up" "No" "Up" ...
## $ glyburide.metformin     : chr  "No" "No" "No" "No" ...
## $ glipizide.metformin     : chr  "No" "No" "No" "No" ...
## $ glimepiride.pioglitazone: chr  "No" "No" "No" "No" ...
## $ metformin.rosiglitazone : chr  "No" "No" "No" "No" ...
## $ metformin.pioglitazone  : chr  "No" "No" "No" "No" ...
## $ change                  : chr  "No" "Ch" "No" "Ch" ...
## $ diabetesMed             : chr  "No" "Yes" "Yes" "Yes" ...
## $ readmitted              : chr  "NO" ">30" "NO" "NO" ...
```

Dataframe “diabetes” is having 101766 rows or instances and each instance has 49 attributes, the target variable is “readmitted” at the end.

All Library Imports

```
library(tidyverse)
library(RColorBrewer)
library(dlookr)
library(ggcorrplot)
library(plyr)
library(dplyr)
library(cowplot)
```

1. Checking the distribution of target class (readmitted) :

```
table(diabetes$readmitted)

##   <30   >30    NO
## 11357 35545 54864


table(diabetes$readmitted)/nrow(diabetes)*100

##      <30      >30       NO
## 11.15992 34.92817 53.91192
```

We can see that the data is target class “readmitted” is imbalanced.
The first table shows the count of data points for each of the 3 classes.
The second table shows the percentage distribution.

The bar plot showing the distribution:

```
ggplot(diabetes, aes(x = readmitted,fill = readmitted)) +
  geom_bar() +
  ggtitle("Distribution of Target Class") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5)) +
  geom_text(aes(label = ..count..), stat = "count", vjust=-0.25) +
  scale_fill_brewer(palette="Pastel1")
```

2. Exploring- encounter_id & patient_nbr:

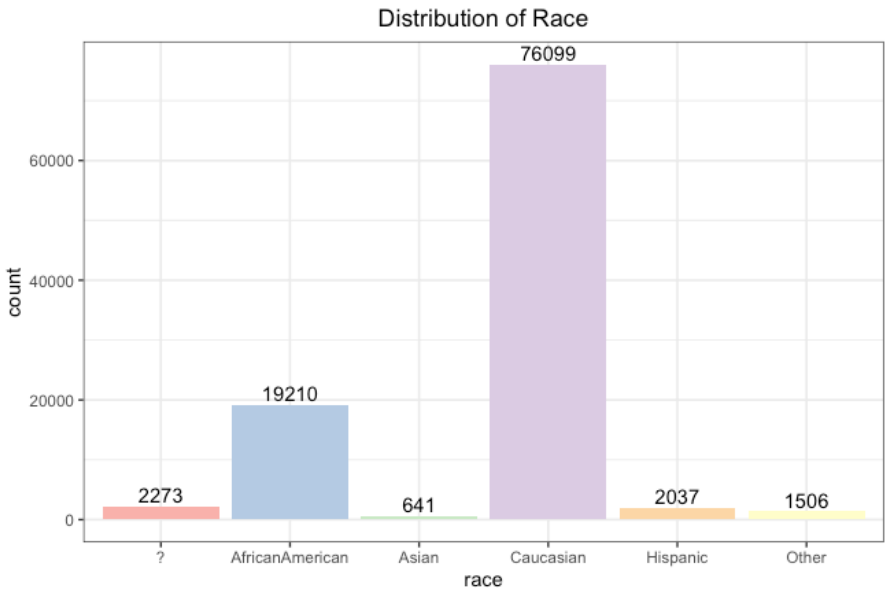
```
count(diabetes)
##        n
## 1 101766

n_distinct(diabetes$encounter_id)
## [1] 101766

n_distinct(diabetes$patient_nbr)
## [1] 71518
```

Unique *encounter_id* are same in count as total number of observations.
This means *encounter_id* can be considered as a primary key of this dataset.

Unique *patient_nbr* is less in count (~71K) , so a patient can have multiple *encounter_id.*
Or a patient was encountered multiple times.

3. Exploring- race: (Categorical)

```
ggplot(diabetes, aes(x = race,fill = race)) +
  geom_bar(show.legend = FALSE) +
  ggtitle("Distribution of Race") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5)) +
  geom_text(aes(label = ..count..), stat = "count", vjust=-0.25) +
  scale_fill_brewer(palette="Pastel1")
```



- Clearly, the “Cacucasian” race dominates.
- The “?” value represents the missing data.

Checking the percentage of missing values

```
count(filter(diabetes, race == "?"))/count(diabetes)*100

##        n
## 1 2.233555
```

Since, since the percentage is quite low, we can keep this attribute.
Hypothesis Test

Checking if “race” is affecting the readmittance:

Applying *Chi-squared Test* on “race” and “readmitted”

```
c_test <- chisq.test(table(diabetes$readmitted, diabetes$race))
c_test

##
##         Pearson's Chi-squared test
##
## data:  table(diabetes$readmitted, diabetes$race)
## X-squared = 282.59, df = 10, p-value < 2.2e-16

c_test$p.value
## [1] 7.379469e-55
```

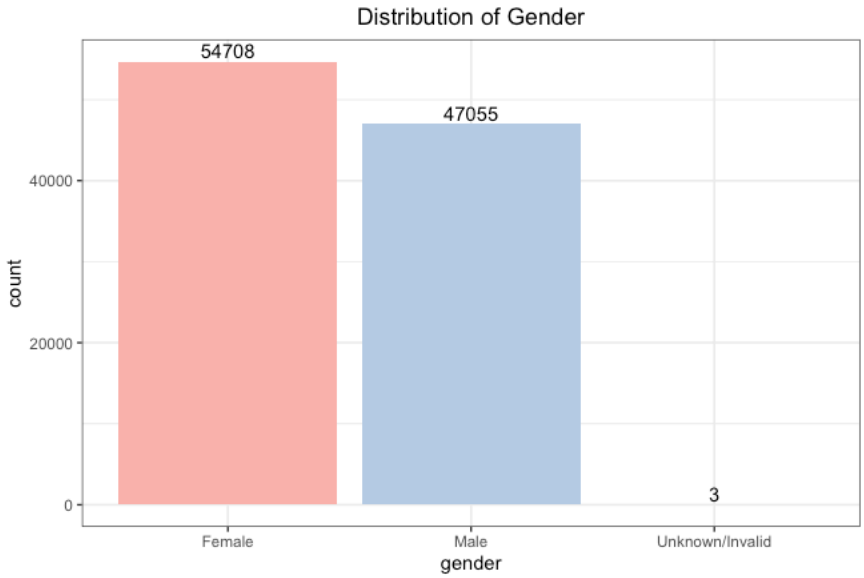
p-value = 7.379469e-55

Since the p-value is much less then 0.05, thus our null hypothesis fails.

This shows that there is a relation between “race” and “readmitted”, and this will be an important attribute for our model.

4. Exploring- gender: (Categorical)

```
ggplot(diabetes, aes(x = gender,fill = gender)) +
  geom_bar(show.legend = FALSE) +
  ggtitle("Distribution of Gender") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5)) +
  geom_text(aes(label = ..count..), stat = "count", vjust=-0.25) +
  scale_fill_brewer(palette="Pastel1")
```



- There is nearly equal distribution of Male and Female genders.
- The “Unknown/Invalid” value represents the missing data, which is quite less (3).

Hypothesis Test

Checking if “gender” is affecting the readmittance:

Applying *Chi-squared Test* on “gender” and “readmitted”

```
## Removing Unknown/Invalid values from gender

daibetes_gen_readm <- filter(diabetes, gender != "Unknown/Invalid" ) %>%
  select(gender, readmitted)


c_test <- chisq.test(table(daibetes_gen_readm$readmitted, daibetes_gen_readm$gender))
c_test

##
##         Pearson's Chi-squared test
##
## data:  table(daibetes_gen_readm$readmitted, daibetes_gen_readm$gender)
## X-squared = 34.896, df = 2, p-value = 2.645e-08

c_test$p.value
## [1] 2.644676e-08
```

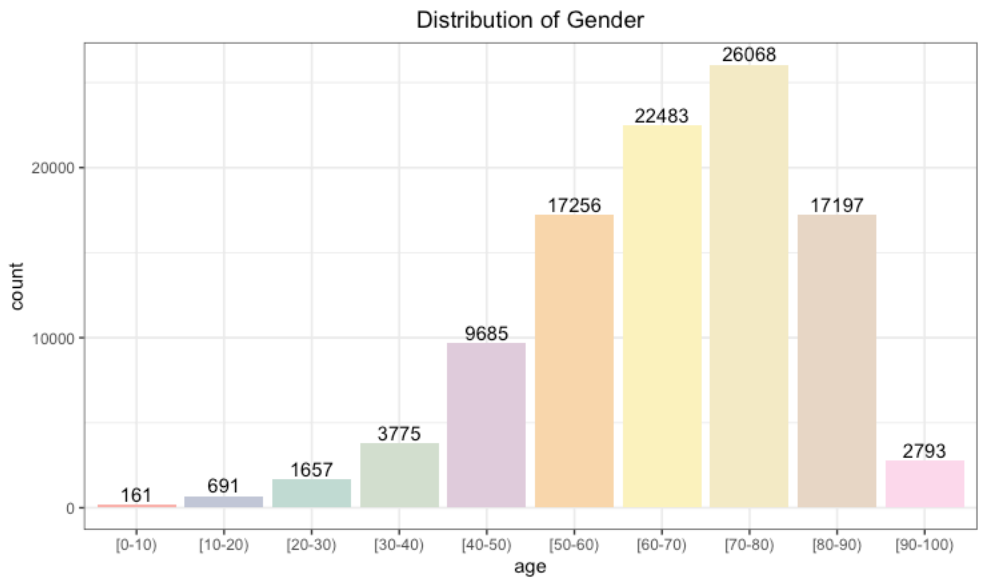
p-value = 2.644676e-08

Since the p-value is much less then 0.05, thus our null hypothesis fails.

This shows that there is a relation between “gender” and “readmitted”, and this will be an important attribute for our model.

5. Exploring- age: (Categorical)

```
## Increasing the color palette
mycolors <- colorRampPalette(brewer.pal(8, "Pastel1"))(10)

ggplot(diabetes, aes(x = age,fill = age)) +
  geom_bar(show.legend = FALSE) +
  ggtitle("Distribution of Gender") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5)) +
  geom_text(aes(label = ..count..), stat = "count", vjust=-0.25) +
  scale_fill_manual(values = mycolors)
```



- People ranging from age 40 to 90 are high in number as compared to other ages.
- This attribute can be used as both Categorical as well as numerical.

Hypothesis Test (if used as Categorical)

Checking if “age” is affecting the readmittance:

Applying *Chi-squared Test* on “age” and “readmitted”

```
c_test <- chisq.test(table(diabetes$readmitted, diabetes$age))
c_test

##        Pearson's Chi-squared test
##
## data:  table(diabetes$readmitted, diabetes$age)
## X-squared = 313.17, df = 18, p-value < 2.2e-16

c_test$p.value

## [1] 9.348415e-56
```

p-value = 9.348415e-56

Since the p-value is much less then 0.05, thus our null hypothesis fails.

This shows that there is a relation between “age” and “readmitted”, and this will be an important attribute for our model.

Hypothesis Test (if used as Numerical)

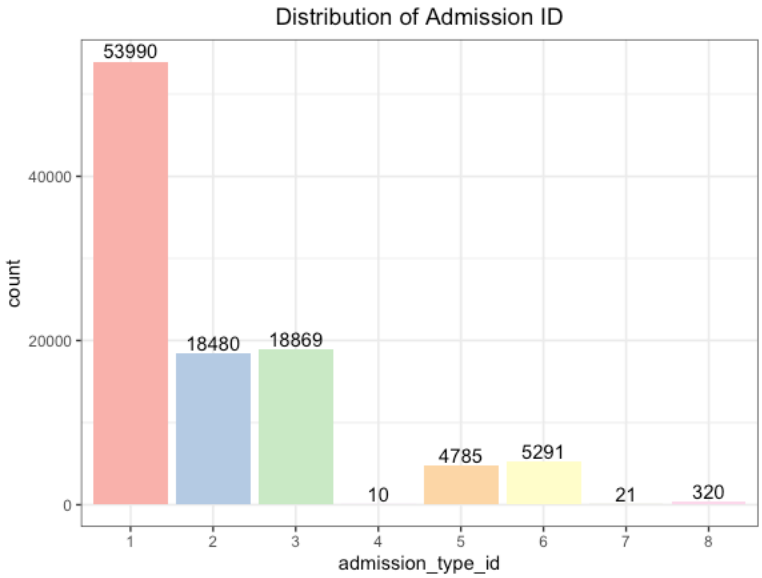
6. Exploring- weight: (Numeric)

```
count(filter(diabetes, weight == "?"))/count(diabetes)*100
##          n
## 1 96.85848
```

Since the percentage of values that are missing for weight attribute is greater than 96.8%, we will **not** use this variable during modeling.

7. Exploring- admission_type_id: (Categorical)

```
new_admission_id <- transform(diabetes, admission_type_id =
as.character(admission_type_id)) %>%
  select(admission_type_id,readmitted)

ggplot(new_admission_id, aes(x = admission_type_id, fill = admission_type_id)) +
  geom_bar(show.legend = FALSE) +
  ggtitle("Distribution of Admission ID") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5)) +
  geom_text(aes(label = ..count..), stat = "count", vjust=-0.25) +
  scale_fill_brewer(palette="Pastel1")
```



admission_type_id	description
1	Emergency
2	Urgent
3	Elective
4	Newborn
5	Not Available
6	NULL
7	Trauma Center
8	Not Mapped

The distribution shows that Emergency entries dominates the entire category.

Hypothesis Test

```
c_test <- chisq.test(table(new_admission_id$readmitted,
new_admission_id$admission_type_id))

c_test

##       Pearson's Chi-squared test
##
## data:  table(new_admission_id$readmitted, new_admission_id$admission_type_id)
## X-squared = 415.76, df = 14, p-value < 2.2e-16

c_test$p.value
## [1] 6.037493e-80
```

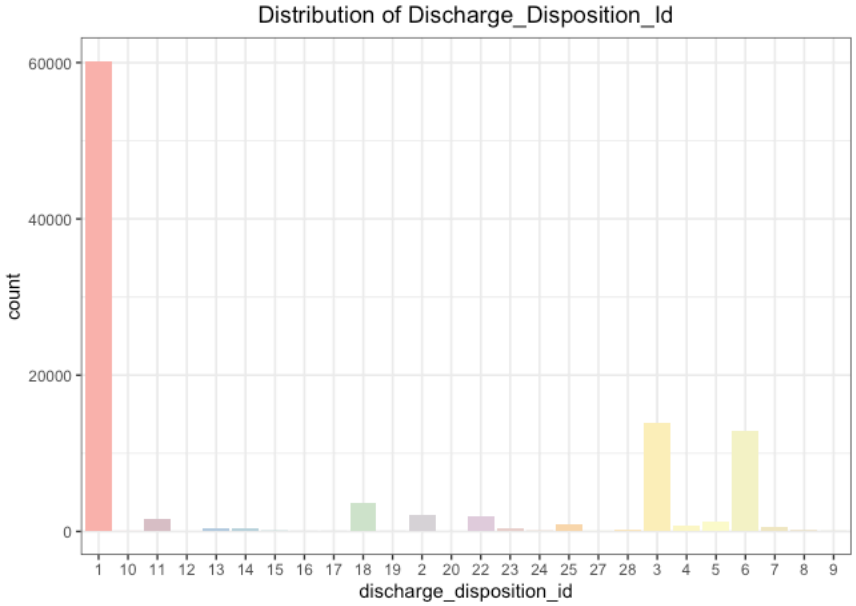
p-value = 6.037493e-80

Since the p-value is much less then 0.05, thus our null hypothesis fails.

This shows that there is a relation between “admission_type_id” and “readmitted”, and this will be an important attribute for our model.

8. Exploring- discharge_disposition_id: (Categorical)

```
diabetes_dispo_id <- transform(diabetes, discharge_disposition_id =
as.character(discharge_disposition_id)) %>%
  select(discharge_disposition_id,readmitted)

mycolors <- colorRampPalette(brewer.pal(8, "Pastel1"))(30)

ggplot(diabetes_dispo_id, aes(x = discharge_disposition_id, fill =
discharge_disposition_id)) +
  geom_bar(show.legend = FALSE) +
  ggtitle("Distribution of Discharge_Disposition_Id") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_fill_manual(values = mycolors)
```



discharge_disposition_id	description
1	Discharged to home
2	Discharged/transferred to another short term hospital
3	Discharged/transferred to SNF
4	Discharged/transferred to ICF
5	Discharged/transferred to another type of inpatient care institution
6	Discharged/transferred to home with home health service
7	Left AMA
8	Discharged/transferred to home under care of Home IV provider
9	Admitted as an inpatient to this hospital
10	Neonate discharged to another hospital for neonatal aftercare
11	Expired
12	Still patient or expected to return for outpatient services
13	Hospice / home
14	Hospice / medical facility
15	Discharged/transferred within this institution to Medicare approved swing bed
16	Discharged/transferred/referred another institution for outpatient services
17	Discharged/transferred/referred to this institution for outpatient services
18	NULL
19	Expired at home. Medicaid only, hospice.
20	Expired in a medical facility. Medicaid only, hospice.
21	Expired, place unknown. Medicaid only, hospice.
22	Discharged/transferred to another rehab fac including rehab units of a hospital .
23	Discharged/transferred to a long term care hospital.
24	Discharged/transferred to a nursing facility certified under Medicaid but not certified under Medicare.
25	Not Mapped
26	Unknown/Invalid
30	Discharged/transferred to another Type of Health Care Institution not Defined Elsewhere
27	Discharged/transferred to a federal health care facility.
28	Discharged/transferred/referred to a psychiatric hospital of psychiatric distinct part unit of a hospital
29	Discharged/transferred to a Critical Access Hospital (CAH).

Hypothesis Test

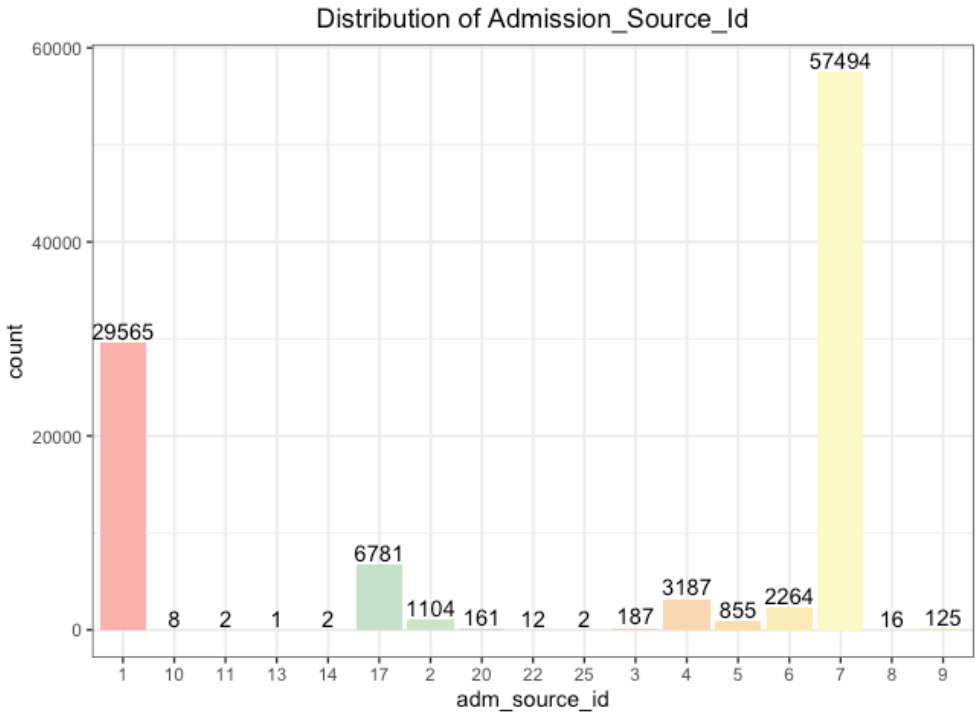
```
c_test <- chisq.test(table(diabetes_dispo_id$readmitted,
diabetes_dispo_id$discharge_disposition_id))

c_test

##        Pearson's Chi-squared test
##
## data:  table(diabetes_dispo_id$readmitted,
## diabetes_dispo_id$discharge_disposition_id)
## X-squared = 3587.3, df = 50, p-value < 2.2e-16

c_test$p.value
## [1] 0
```

p-value = 0

From the figure and p values, we can see that this categorical variable is heavily skewed.

9. Exploring- admission_source_id: (Categorical)

```
diabetes_adm_source <- transform(diabetes, adm_source_id =
as.character(admission_source_id)) %>%
  select(adm_source_id,readmitted)

mycolors <- colorRampPalette(brewer.pal(8, "Pastel1"))(21)

ggplot(diabetes_adm_source, aes(x = adm_source_id, fill = adm_source_id)) +
  geom_bar(show.legend = FALSE) +
  ggtitle("Distribution of Admission_Source_Id") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5)) +
  geom_text(aes(label = ..count..), stat = "count", vjust=-0.25) +
  scale_fill_manual(values = mycolors)
```



Hypothesis Test

```
c_test <- chisq.test(table(diabetes_adm_source$readmitted,
diabetes_adm_source$adm_source_id))

c_test

##        Pearson's Chi-squared test
##
## data:  table(diabetes_adm_source$readmitted, diabetes_adm_source$adm_source_id)
## X-squared = 1151, df = 32, p-value < 2.2e-16

c_test$p.value

## [1] 2.317985e-221
```

p-value = 2.317985e-221

From the figure and p values, we can see that this categorical variable is heavily skewed.

admission_source_id	description
1	Physician Referral
2	Clinic Referral
3	HMO Referral
4	Transfer from a hospital
5	Transfer from a Skilled Nursing Facility (SNF)
6	Transfer from another health care facility
7	Emergency Room
8	Court/Law Enforcement
9	Not Available
10	Transfer from critial access hospital
11	Normal Delivery
12	Premature Delivery
13	Sick Baby
14	Extramural Birth
15	Not Available
17	NULL
18	Transfer From Another Home Health Agency
19	Readmission to Same Home Health Agency
20	Not Mapped
21	Unknown/Invalid
22	Transfer from hospital inpt/same fac reslt in a sep claim
23	Born inside this hospital
24	Born outside this hospital
25	Transfer from Ambulatory Surgery Center
26	Transfer from Hospice

Emergency Room tops the bar graph and the next is Physician Referral.

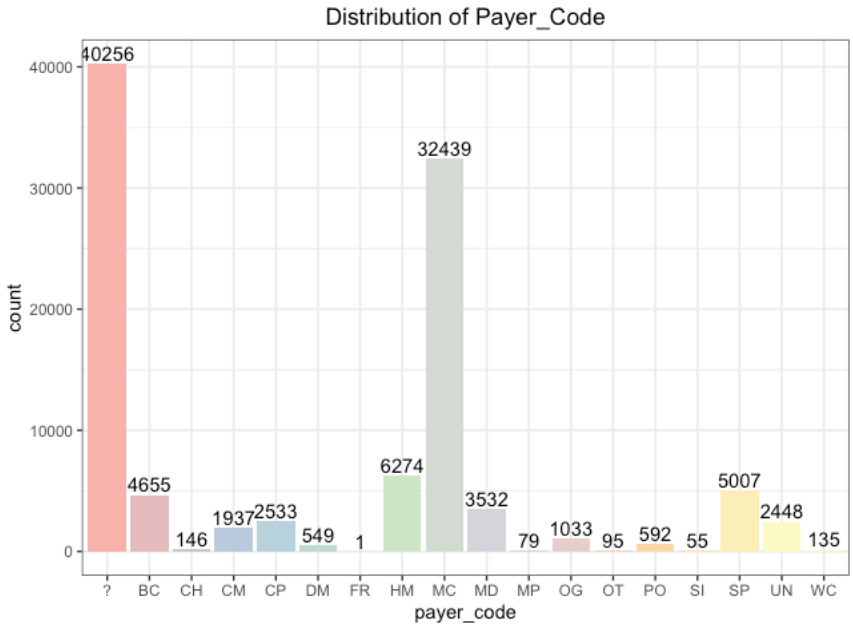
10. Exploring- payer_code: (Categorical)

```
count(filter(diabetes, payer_code == "?"))/count(diabetes)*100
##          n
## 1 39.55742
```

Nearly 40% of the values are missing for this attribute.

If we treat this as a separate category, let's see how it performs on hypothesis tests.

```
mycolors <- colorRampPalette(brewer.pal(8, "Pastel1"))(24)

ggplot(diabetes, aes(x = payer_code, fill = payer_code)) +
  geom_bar(show.legend = FALSE) +
  ggtitle("Distribution of Payer_Code") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5)) +
  geom_text(aes(label = ..count..), stat = "count", vjust=-0.25) +
  scale_fill_manual(values = mycolors)
```




```
c_test <- chisq.test(table(diabetes$readmitted, diabetes$payer_code))
c_test

##        Pearson's Chi-squared test
##
## data:  table(diabetes$readmitted, diabetes$payer_code)
## X-squared = 521.16, df = 34, p-value < 2.2e-16

c_test$p.value
## [1] 1.559345e-88
```

p-value = 1.559345e-88

If we treat “?” that is unknown as a separate category, then the p-value < 0.05, which means this attribute is important for predicting our target.

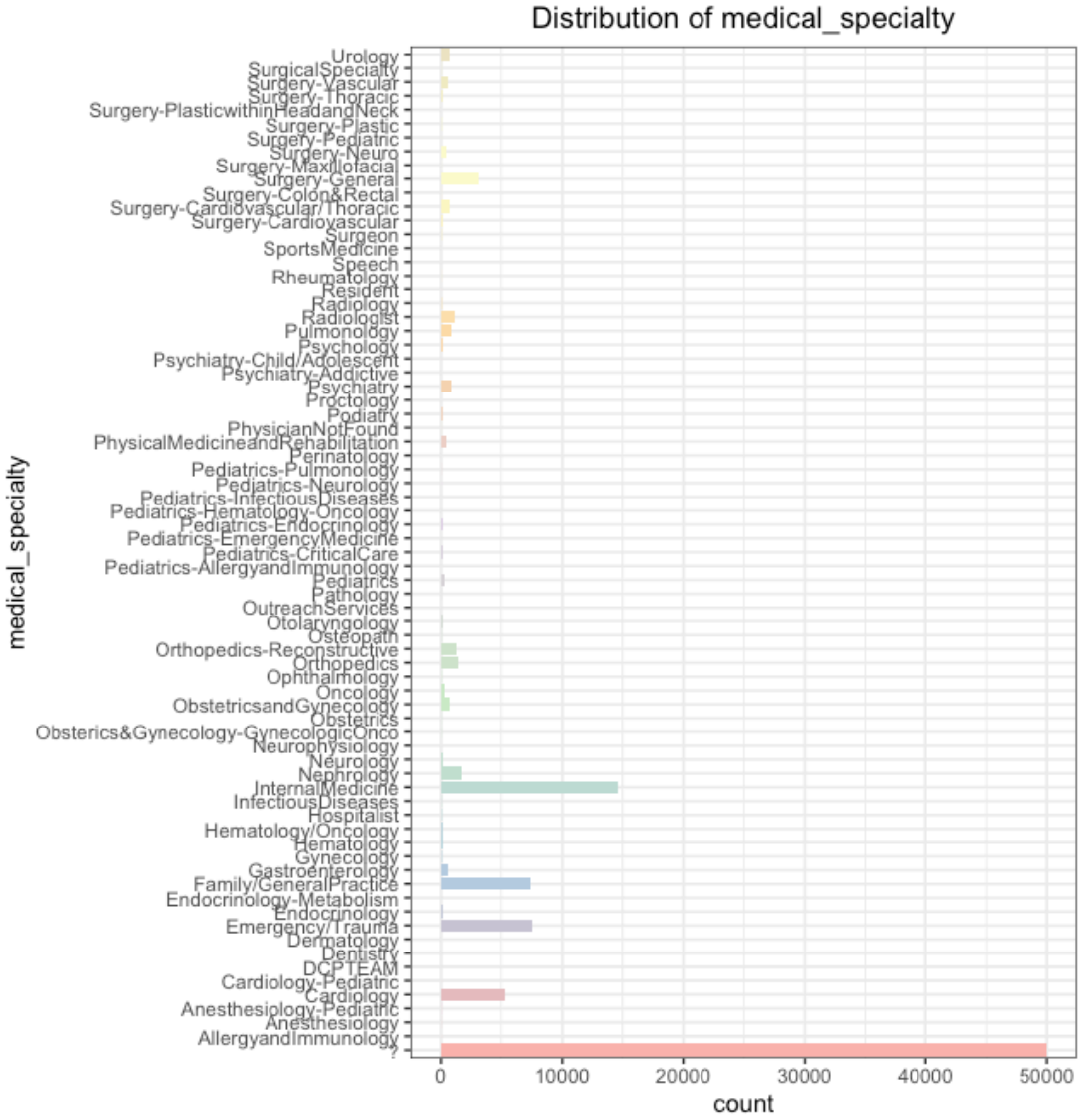
11. Exploring- medical_specialty: (Categorical)

```
count(filter(diabetes, medical_specialty == "?"))/count(diabetes)*100

##          n
## 1 49.08221
```

49% of the values are missing for this attribute. (We can drop it, but still will try once)

If we treat this as a separate category, let’s see how it performs on hypothesis tests.

```
mycolors <- colorRampPalette(brewer.pal(8, "Pastel1"))(90)

ggplot(diabetes, aes(x = medical_specialty, fill = medical_specialty)) +
  geom_bar(show.legend = FALSE) +
  ggtitle("Distribution of medical_specialty") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_fill_manual(values = mycolors) +
  coord_flip()
```



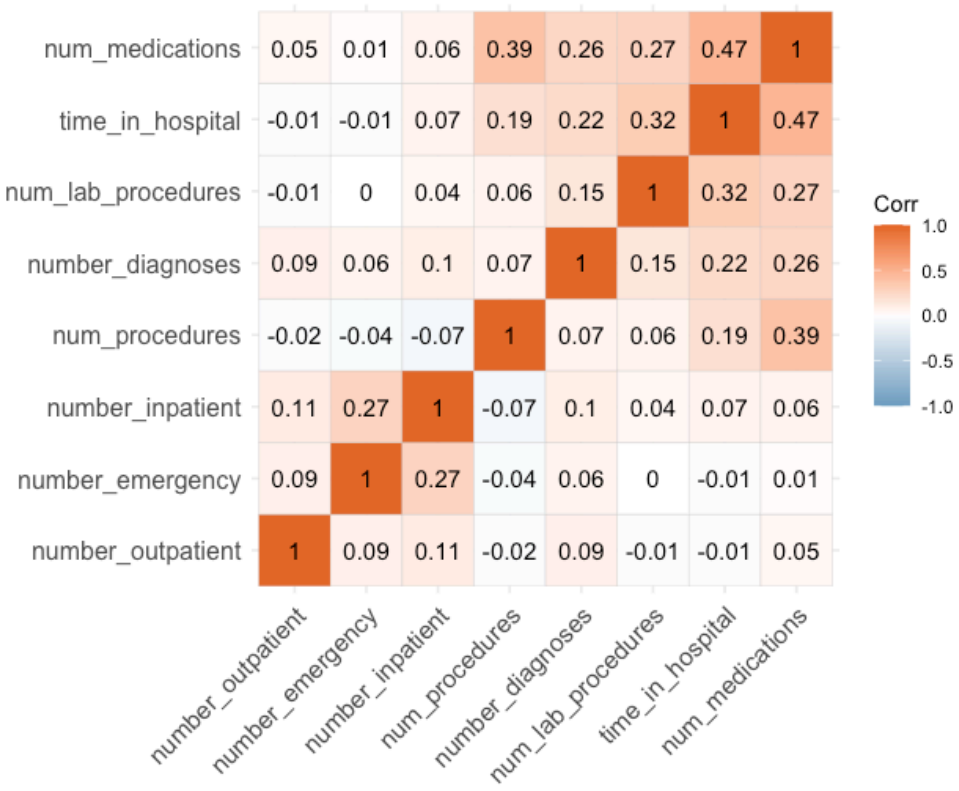

```
c_test <- c_test <- chisq.test(table(diabetes$readmitted,
diabetes$medical_specialty))

c_test

##        Pearson's Chi-squared test
##
## data:  table(diabetes$readmitted, diabetes$medical_specialty)
## X-squared = 1354.6, df = 144, p-value < 2.2e-16

c_test$p.value
## [1] 9.213296e-196
```

p-value = 9.213296e-196

The p-value is nearly 0, but we need to perform a lot of pre-processing, before using this attribute.

12. Correlation - Numeric Variables:

```
library(ggcorrplot)

numeric = c("time_in_hospital",
            "num_lab_procedures",
            "num_procedures",
            "num_medications",
            "number_outpatient",
            "number_emergency",
            "number_inpatient",
            "number_diagnoses")

diabetes_numeric <- select(diabetes, numeric)

correlation_matrix <- round(cor(diabetes_numeric),2)

ggcorrplot(correlation_matrix, hc.order = TRUE, lab = TRUE,
           colors = c("#6D9EC1", "white", "#E46726"))
```

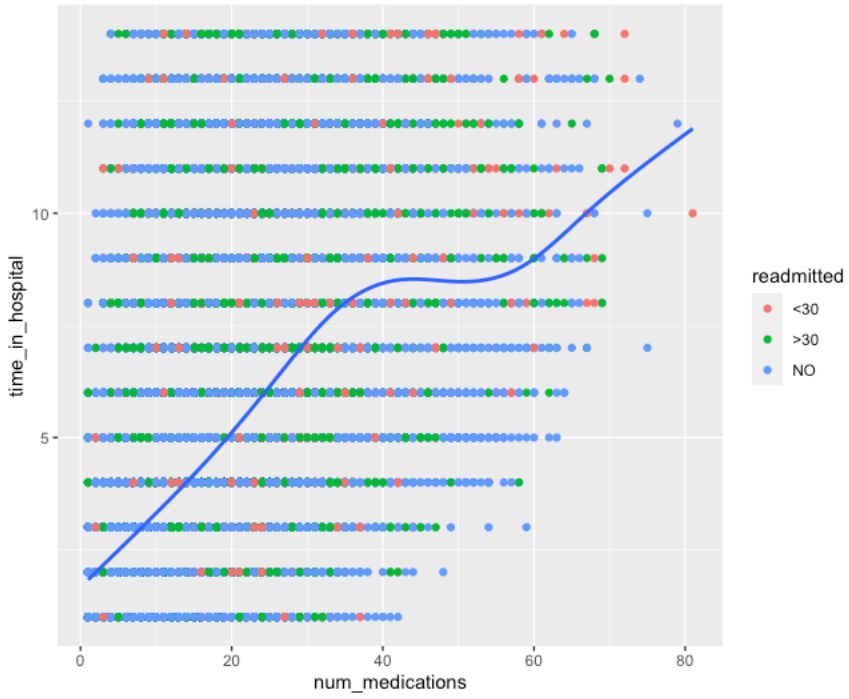


- num_medications & time_in_hospital are moderately correlated (positive).
- Next is num_medications & num_procedures
- Next, time_in_hospital & num_lab_procedures
- Next, number_inpatient & number_emergency

13. Relational Plots between correlated attributes:

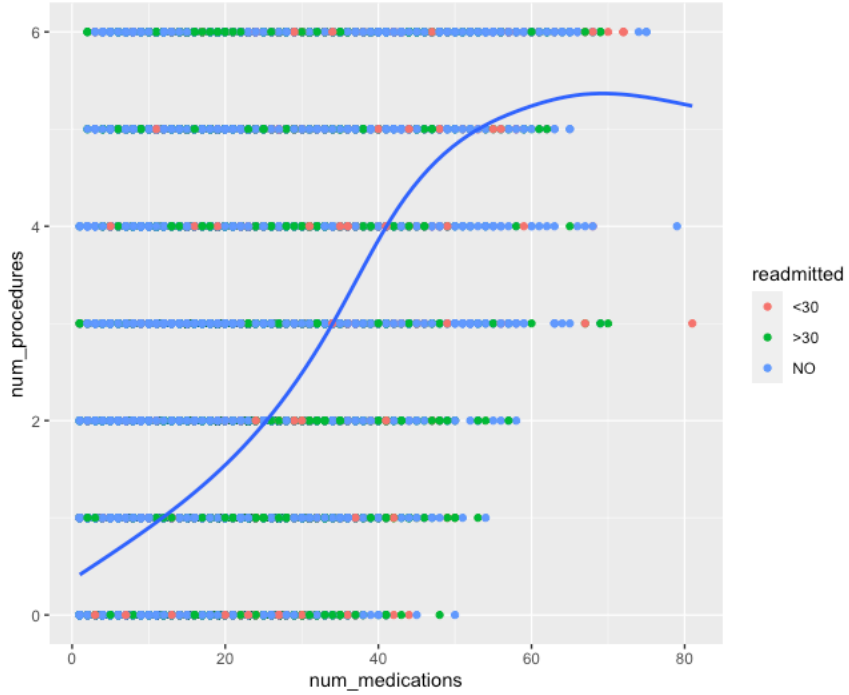
num_medications and time_in_hospital (correlation = 0.47)

```
ggplot(data = diabetes, mapping = aes(x = num_medications, y = time_in_hospital)) +
  geom_point(mapping = aes(color=readmitted)) +
  geom_smooth(se = FALSE)
```



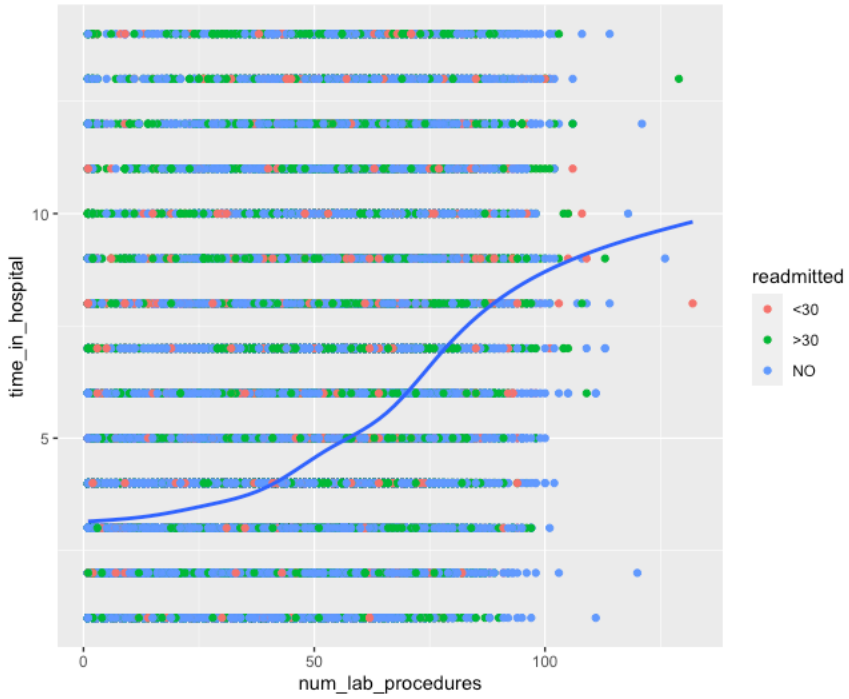
num_medications and num_procedures (correlation = 0.39)

```
ggplot(data = diabetes, mapping = aes(x = num_medications, y = num_procedures)) +
  geom_point(mapping = aes(color=readmitted)) +
  geom_smooth(se = FALSE)
```



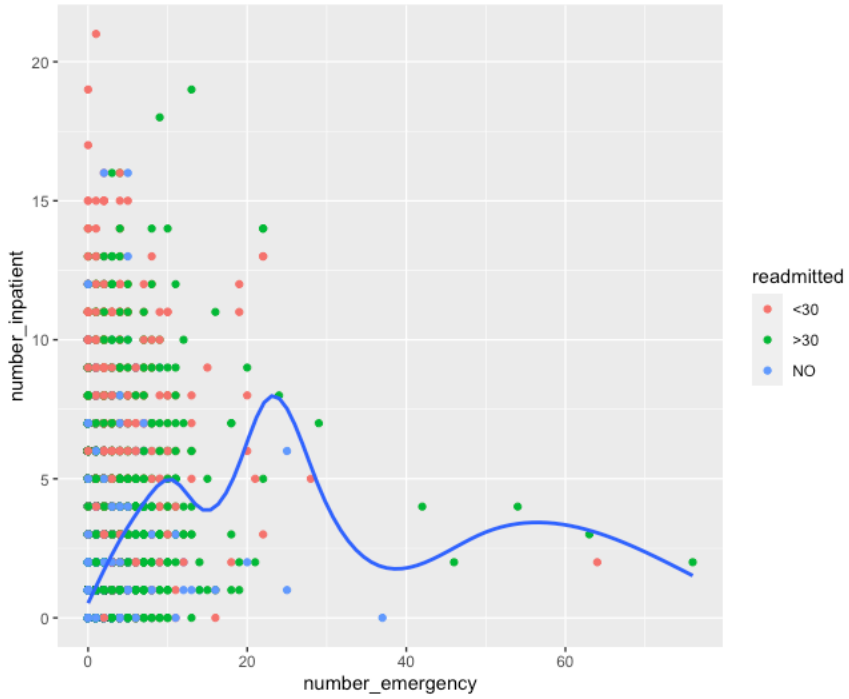
num_lab_procedures and time_in_hospital (correlation = 0.32)

```
ggplot(data = diabetes, mapping = aes(x = num_lab_procedures,y = time_in_hospital)) +
  geom_point(mapping = aes(color=readmitted)) +
  geom_smooth(se = FALSE)
```



number_emergency and number_inpatient (correlation = 0.27)

```
ggplot(data = diabetes, mapping = aes(x = number_emergency, y = number_inpatient)) +
  geom_point(mapping = aes(color=readmitted)) +
  geom_smooth(se = FALSE)
```



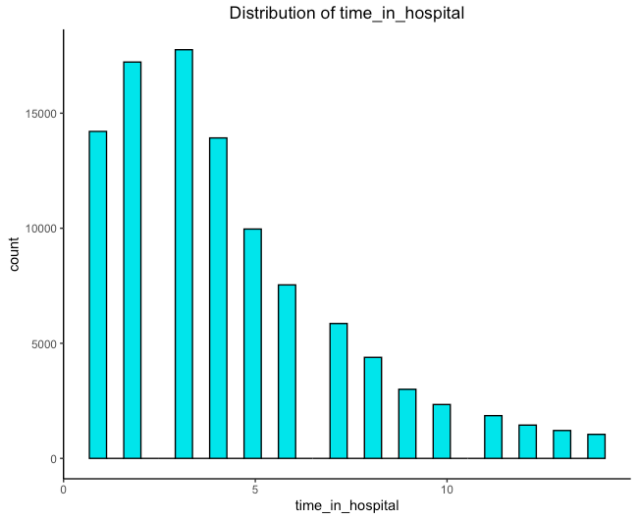
14. Exploring- time_in_hospital: (Numeric)

Counting the null values if any:

```
sum(is.na(diabetes$time_in_hospital))

## [1] 0
```

Checking the distribution

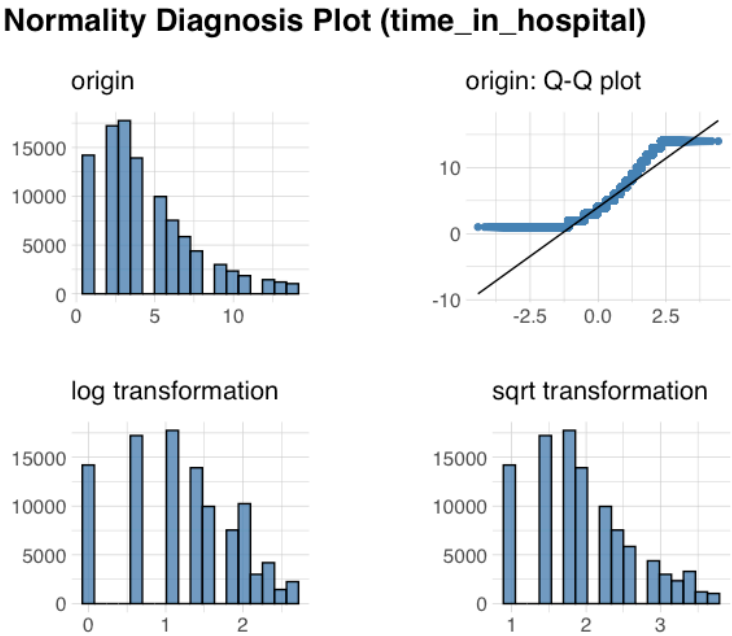
```
ggplot(diabetes, mapping = aes(time_in_hospital)) +
  geom_histogram(fill="turquoise2", color="black") +
  ggtitle("Distribution of time_in_hospital") +
  theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))
```



The histogram plot is right skewed, and clearly doesn't look normally distributed.

Normality Plot:

```
plot_normality(diabetes, time_in_hospital)
```



On the Q-Q plot, the points deviating from the straight line, this shows that the distribution is not normal.
Also, the log and sqrt transforms does not make much difference in the distribution.

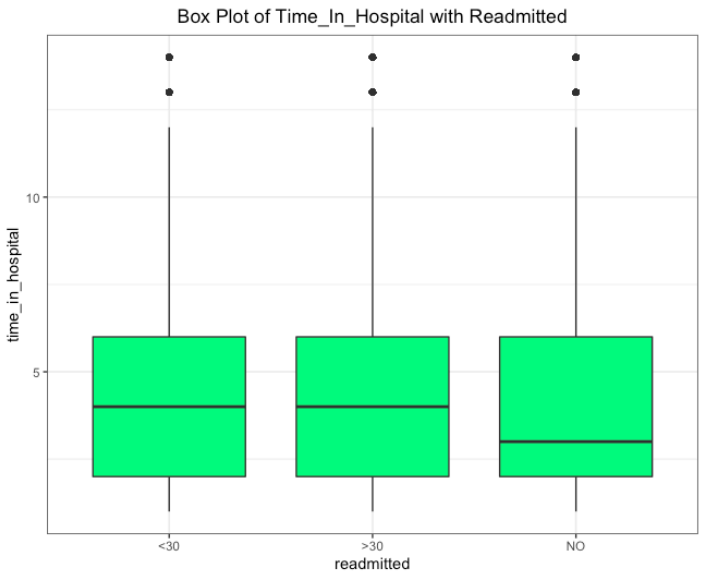
Normality Test:

```
skewness(diabetes$time_in_hospital)

## [1] 1.133982

## Shapiro-Wilk normality test
normality(diabetes, time_in_hospital)

##  # A tibble: 1 × 4
##    vars             statistic  p_value sample
##    <chr>                <dbl>    <dbl>  <dbl>
##  1 time_in_hospital     0.883 6.03e-52   5000
```

Right skewed, as show from skewness.
The information from the plots is now verified, the distribution is not Normal.

Box Plot:

```
ggplot(data = diabetes, mapping = aes(x = readmitted, y = time_in_hospital)) +
  geom_boxplot(fill="springgreen1")+
  ggtitle("Box Plot of Time_In_Hospital with Readmitted") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



Some outliers are present for all the 3 classes.

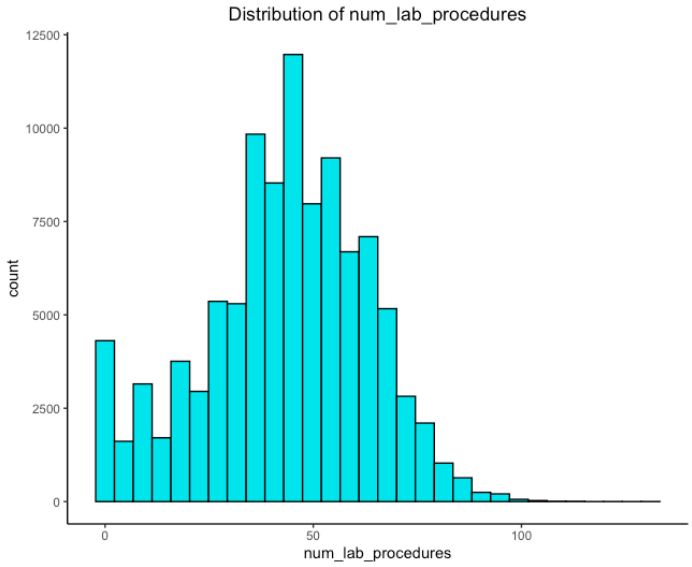
15. Exploring- num_lab_procedures: (Numeric)

Counting the null values if any:

```
sum(is.na(diabetes$num_lab_procedures))

## [1] 0
```

Checking the distribution

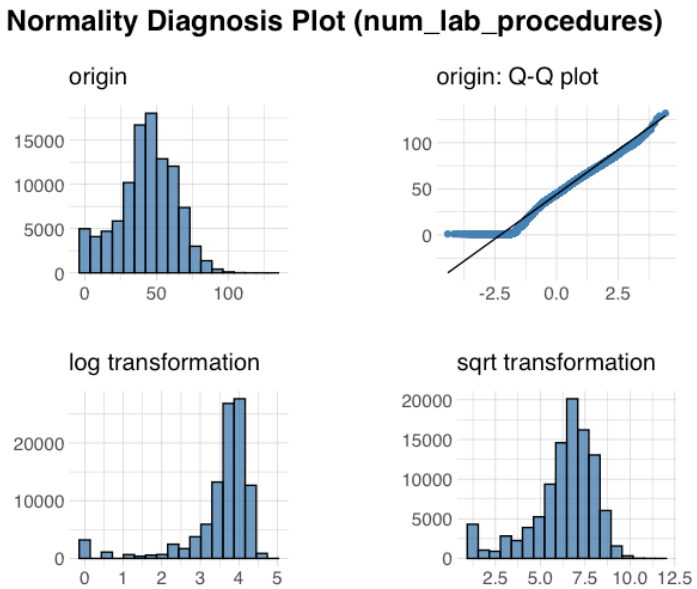
```
ggplot(diabetes, mapping = aes(num_lab_procedures)) +
  geom_histogram(fill="turquoise2", color="black") +
  ggtitle("Distribution of num_lab_procedures") +
  theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))
```



The histogram plot is skewed, and clearly doesn't look normally distributed.

Normality Plot:

```
plot_normality(diabetes, num_lab_procedures)
```



On the Q-Q plot, the points deviating from the straight line, this shows that the distribution is not normal. Also, the log and sqrt transforms does not make much difference in the distribution.

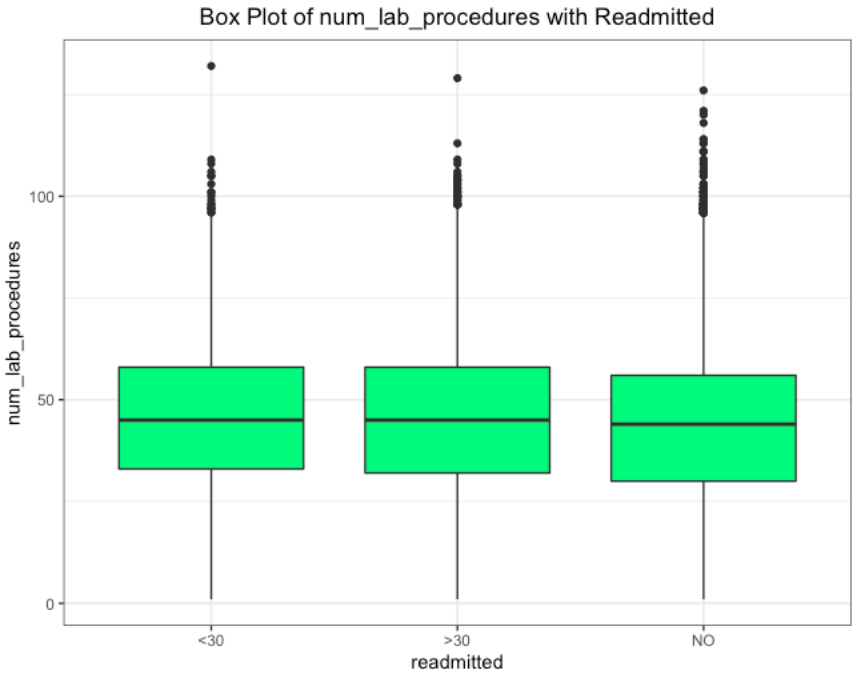
Normality Test:

```
skewness(diabetes$num_lab_procedures)

## [1] -0.2365404

## Shapiro-Wilk normality test
normality(diabetes, num_lab_procedures)

## # A tibble: 1 × 4
##   vars               statistic  p_value sample
##   <chr>                  <dbl>    <dbl>  <dbl>
## 1 num_lab_procedures     0.982 1.33e-24   5000
```

Left skewed, as show from skewness.
The information from the plots is now verified, the distribution is not Normal.

Box Plot:

```
ggplot(data = diabetes, mapping = aes(x = readmitted, y = num_lab_procedures)) +
  geom_boxplot(fill="springgreen1")+
  ggtitle("Box Plot of num_lab_procedures with Readmitted") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



Many outliers are present for all the 3 classes.

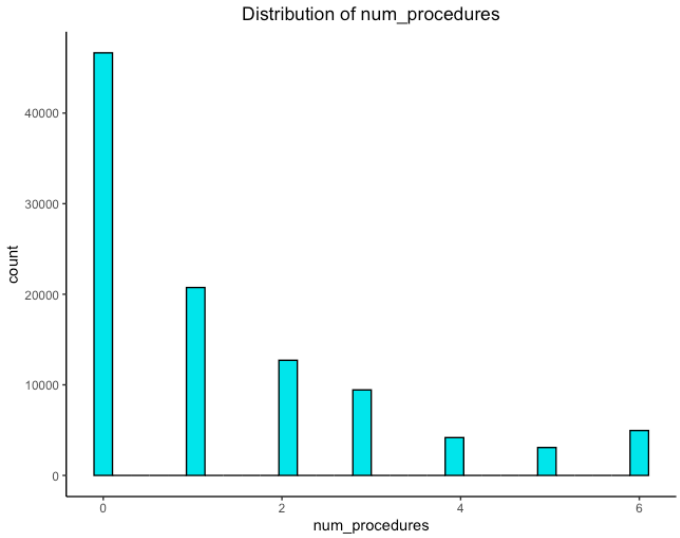
16. Exploring- num_procedures: (Numeric)

Counting the null values if any:

```
sum(is.na(diabetes$num_procedures))

## [1] 0
```

Checking the distribution

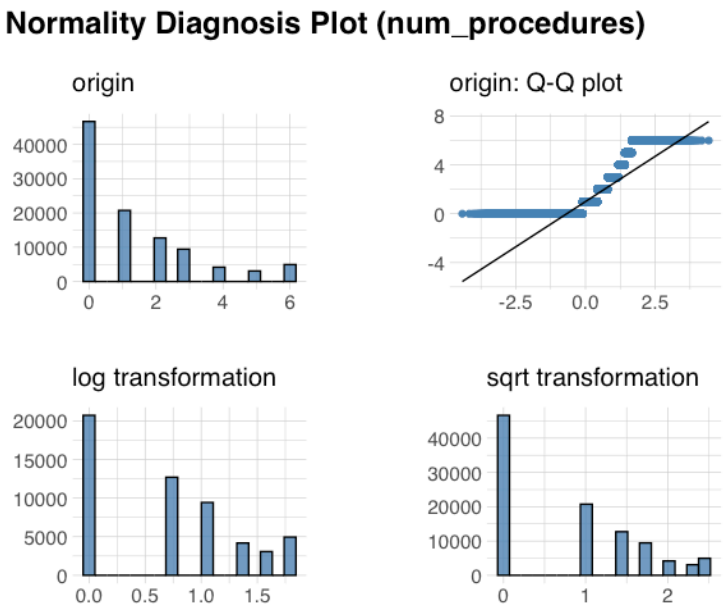
```
ggplot(diabetes, mapping = aes(num_procedures)) +
  geom_histogram(fill="turquoise2", color="black") +
  ggtitle("Distribution of num_procedures") +
  theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))
```



The histogram plot is shows the attribute has few discrete values.

Normality Plot:

```
plot_normality(diabetes, num_lab_procedures)
```



On the Q-Q plot, the points deviating from the straight line, this shows that the distribution is not normal. Also, the log and sqrt transforms does not make much difference in the distribution.

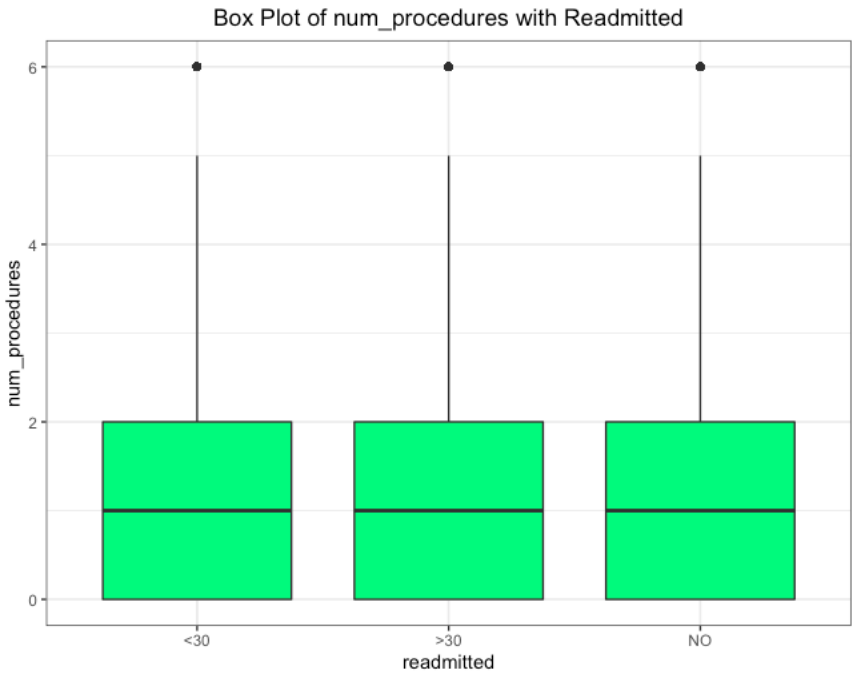
Normality Test:

```
skewness(diabetes$num_procedures)

## [1] 1.316395

## Shapiro-Wilk normality test
normality(diabetes, num_procedures)

## # A tibble: 1 × 4
##   vars           statistic  p_value sample
##   <chr>              <dbl>    <dbl>  <dbl>
## 1 num_procedures     0.772 4.51e-64   5000
```

Left skewed, as show from skewness.
The information from the plots is now verified, the distribution is not Normal.

Box Plot:

```
ggplot(data = diabetes, mapping = aes(x = readmitted, y = num_procedures)) +
  geom_boxplot(fill="springgreen1")+
  ggtitle("Box Plot of num_procedures with Readmitted") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



Some outliers are present for all the 3 classes.

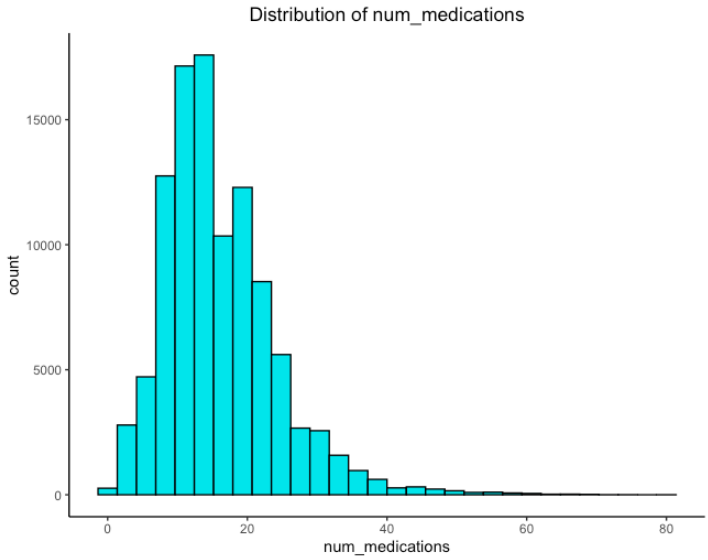
17. Exploring- num_medications: (Numeric)

Counting the null values if any:

```
sum(is.na(diabetes$num_medications))

## [1] 0
```

Checking the distribution

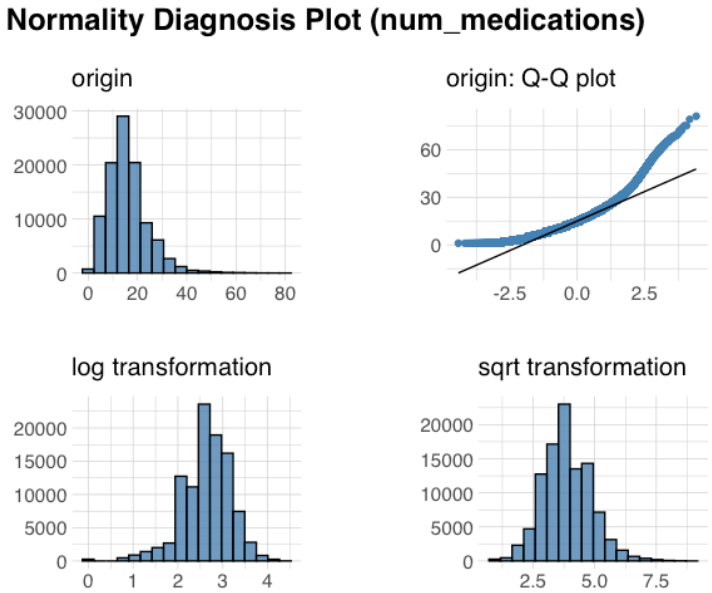
```
ggplot(diabetes, mapping = aes(num_medications)) +
  geom_histogram(fill="turquoise2", color="black") +
  ggtitle("Distribution of num_medications") +
  theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))
```



The histogram plot is skewed, and clearly doesn't look normally distributed.

Normality Plot:

```
plot_normality(diabetes, num_medications)
```



On the Q-Q plot, the points deviating from the straight line, this shows that the distribution is not normal. However, **sqrt transform** does make a difference in the distribution.

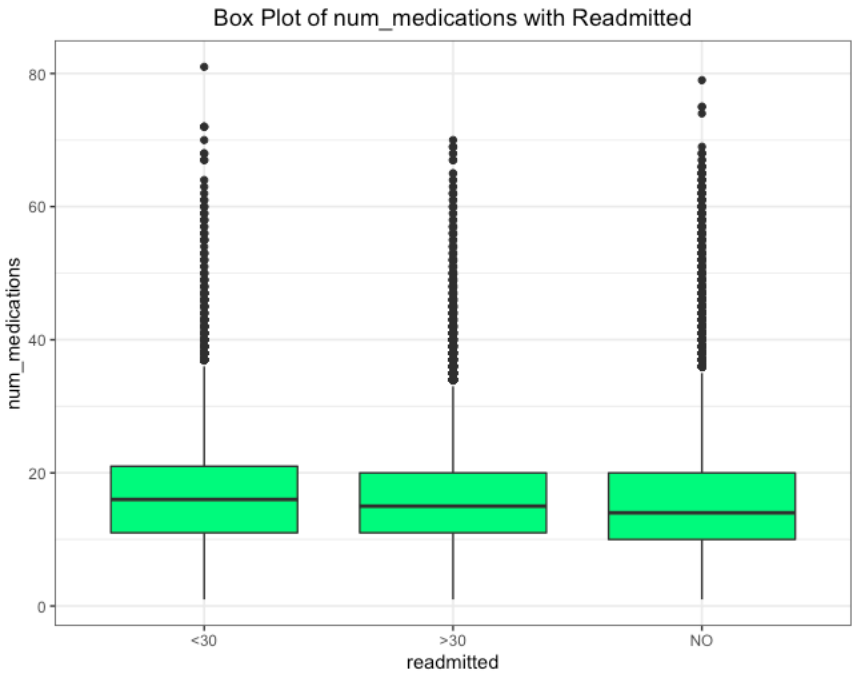
Normality Test:

```
skewness(diabetes$num_medications)

## [1] 1.326653

## Shapiro-Wilk normality test
normality(diabetes, num_medications)

## # A tibble: 1 × 4
##   vars            statistic  p_value sample
##   <chr>               <dbl>    <dbl>  <dbl>
## 1 num_medications     0.921 2.96e-45   5000
```

Right skewed, as show from skewness.
The information from the plots is now verified, the distribution is not Normal.

Box Plot:

```
ggplot(data = diabetes, mapping = aes(x = readmitted, y = num_medications)) +
  geom_boxplot(fill="springgreen1")+
  ggtitle("Box Plot of num_medications with Readmitted") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



A lot of outliers are present for all the 3 classes.

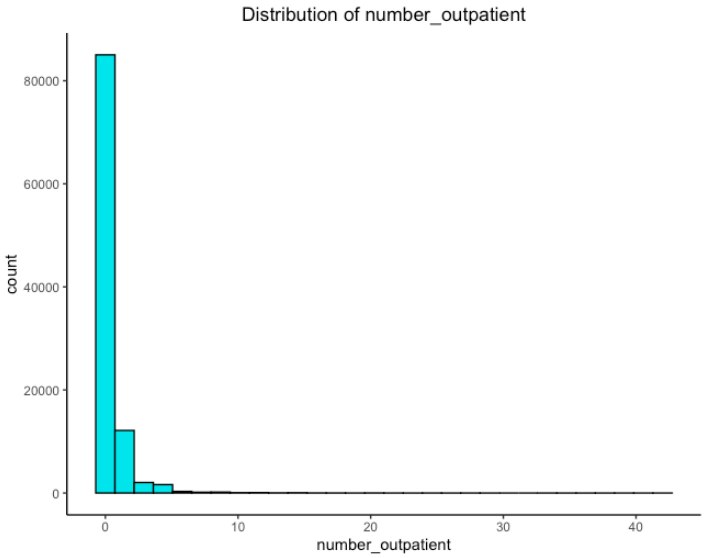
18. Exploring- number_outpatient: (Numeric)

Counting the null values if any:

```
sum(is.na(diabetes$number_outpatient))

## [1] 0
```

Checking the distribution

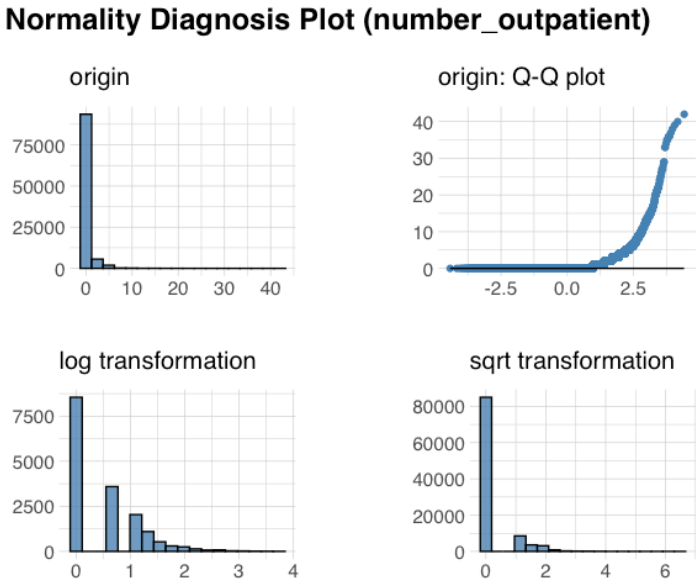
```
ggplot(diabetes, mapping = aes(number_outpatient)) +
  geom_histogram(fill="turquoise2", color="black") +
  ggtitle("Distribution of number_outpatient") +
  theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))
```



The histogram plot is skewed, and clearly doesn't look normally distributed.

Normality Plot:

```
plot_normality(diabetes, number_outpatient)
```



On the Q-Q plot, the points deviating from the straight line, this shows that the distribution is not normal. Also, the log and sqrt transforms does not make much difference in the distribution.

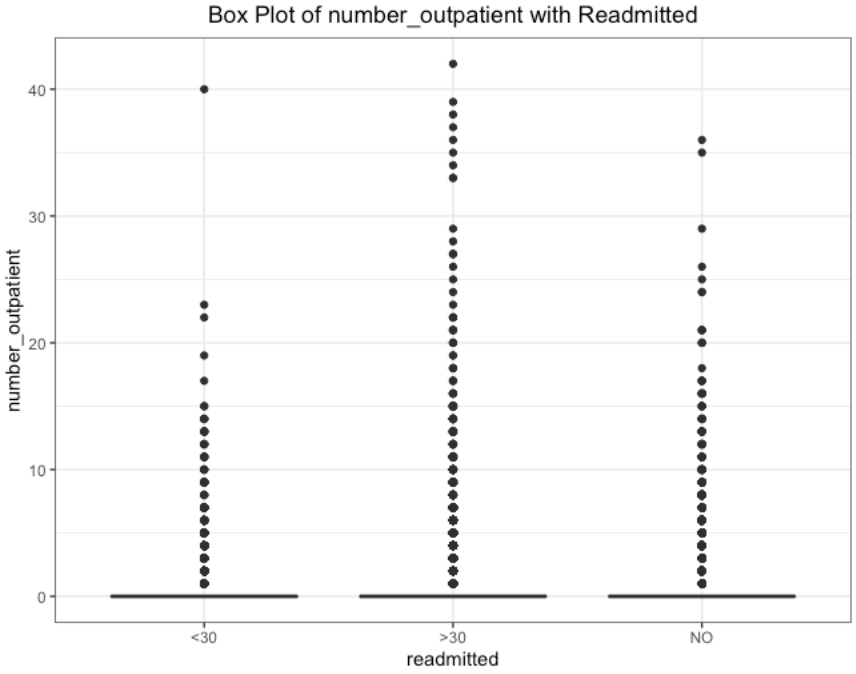
Normality Test:

```
skewness(diabetes$number_outpatient)

## [1] 8.832829

## Shapiro-Wilk normality test
normality(diabetes, number_outpatient)

## # A tibble: 1 × 4
##   vars              statistic  p_value sample
##   <chr>                 <dbl>    <dbl>  <dbl>
## 1 number_outpatient     0.323 7.90e-87   5000
```

Right skewed heavily, as show from skewness.
The information from the plots is now verified, the distribution is not Normal.

Box Plot:

```
ggplot(data = diabetes, mapping = aes(x = readmitted, y = number_outpatient)) +
  geom_boxplot(fill="springgreen1")+
  ggtitle("Box Plot of number_outpatient with Readmitted") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



A lot of outliers are present for all the 3 classes.

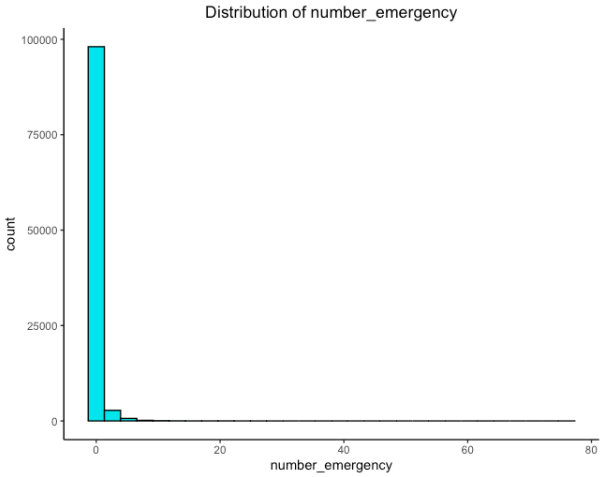
19. Exploring- number_emergency: (Numeric)

Counting the null values if any:

```
sum(is.na(diabetes$number_emergency))

## [1] 0
```

Checking the distribution

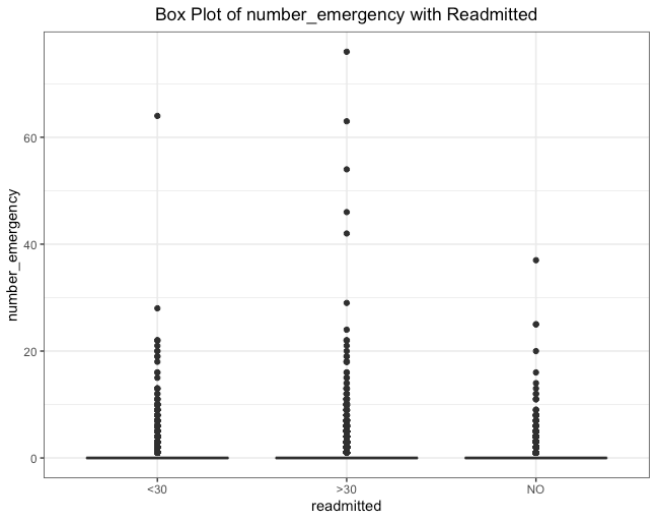
```
ggplot(diabetes, mapping = aes(number_emergency)) +
  geom_histogram(fill="turquoise2", color="black") +
  ggtitle("Distribution of number_emergency") +
  theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))
```



Heavily skewed, not required to check normality.

Also the Box plot will have multiple outliers.

```
ggplot(data = diabetes, mapping = aes(x = readmitted, y = number_emergency)) +
  geom_boxplot(fill="springgreen1")+
  ggtitle("Box Plot of number_emergency with Readmitted") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



A lot of outliers are present for all the 3 classes.

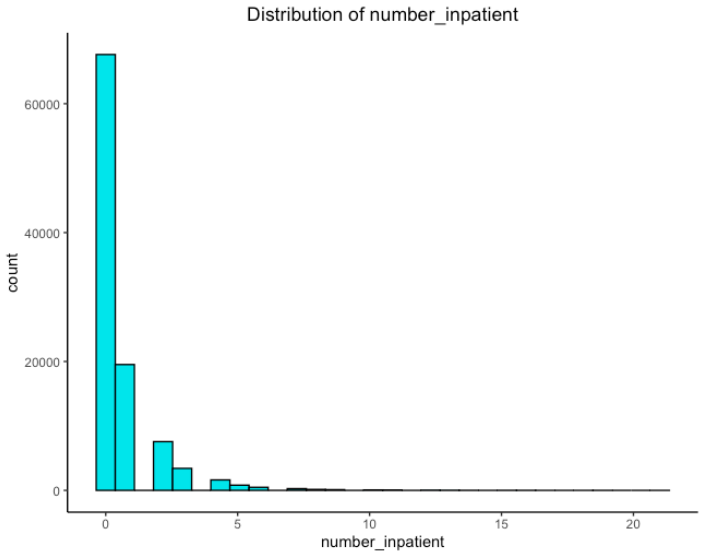
20. Exploring- number_inpatient: (Numeric)

Counting the null values if any:

```
sum(is.na(diabetes$number_inpatient))

## [1] 0
```

Checking the distribution

```
ggplot(diabetes, mapping = aes(number_inpatient)) +
  geom_histogram(fill="turquoise2", color="black") +
  ggtitle("Distribution of number_inpatient") +
  theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))
```



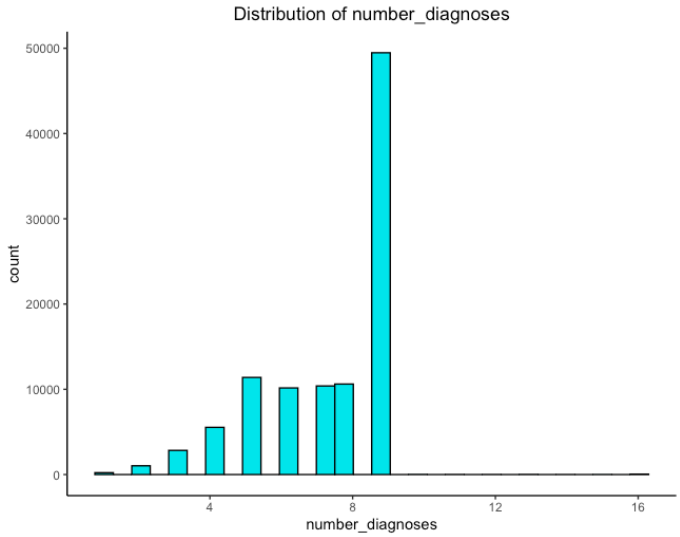
Heavily skewed, not required to check normality.

Also, the Box plot will have multiple outliers.

21. Exploring- number_diagnoses: (Numeric)

Checking the distribution

```
ggplot(diabetes, mapping = aes(number_diagnoses)) +
  geom_histogram(fill="turquoise2", color="black") +
  ggtitle("Distribution of number_diagnoses") +
  theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))
```

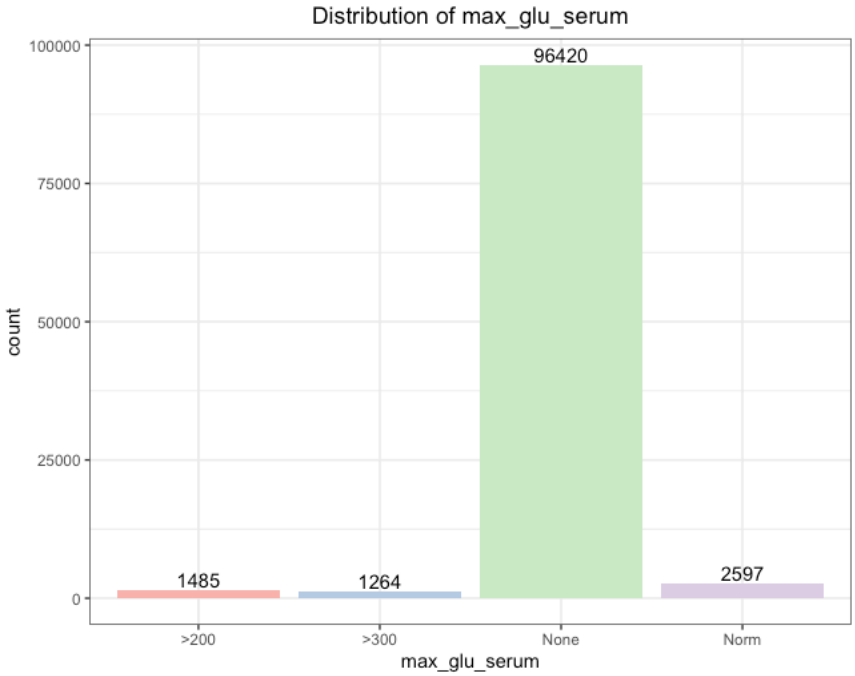


Heavily skewed, not required to check normality.

Also, the Box plot will have multiple outliers.

22. Exploring- max_glu_serum: (Categorical)

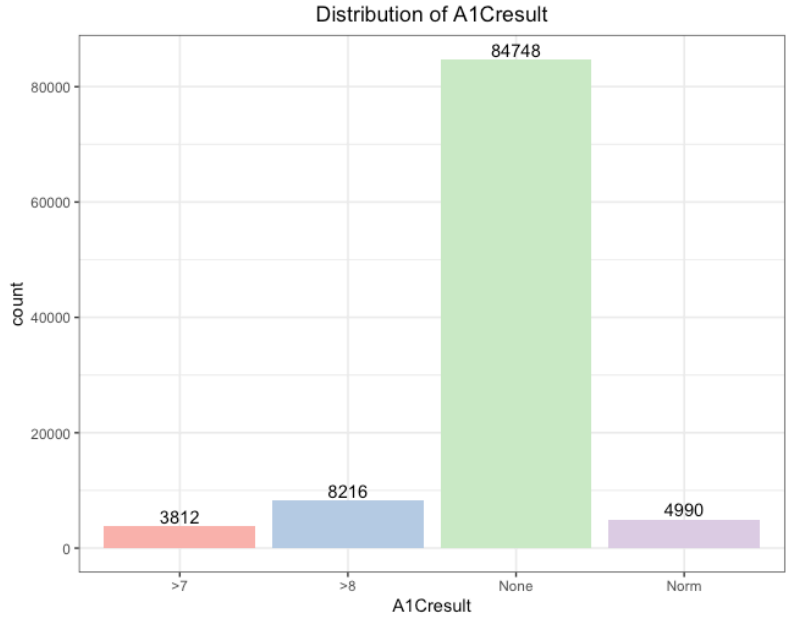
```
ggplot(diabetes, aes(x = max_glu_serum, fill = max_glu_serum)) +
  geom_bar(show.legend = FALSE) +
  ggtitle("Distribution of max_glu_serum") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5)) +
  geom_text(aes(label = ..count..), stat = "count", vjust=-0.25) +
  scale_fill_brewer(palette="Pastel1")
```



Most of the values are None, thus we will drop this variable.

23. Exploring- A1Cresult: (Categorical)

```
ggplot(diabetes, aes(x = A1Cresult, fill = A1Cresult)) +
  geom_bar(show.legend = FALSE) +
  ggtitle("Distribution of A1Cresult") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5)) +
  geom_text(aes(label = ..count..), stat = "count", vjust=-0.25) +
  scale_fill_brewer(palette="Pastel1")
```



Most of the values are None, thus we will drop this variable.

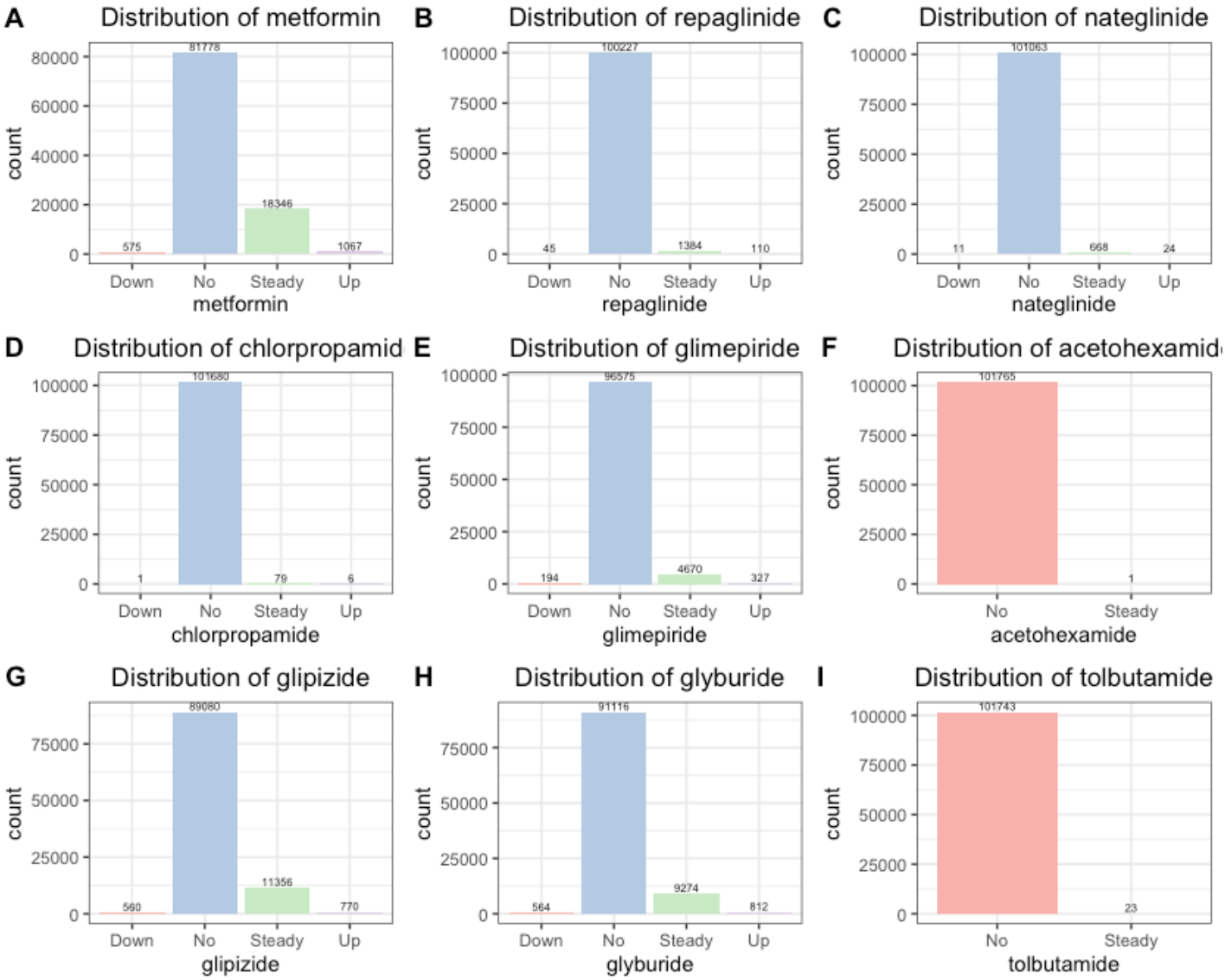
24. Exploring- Features of Medication: (Categorical)

```
## Function to plot distribution of a categorical variable
cat_distribution <- function(df, atr) {
  title <- paste("Distribution of",atr, sep=" ")
  plt <- ggplot(df, aes_string(x = atr, fill = atr)) +
    geom_bar(show.legend = FALSE) +
    ggtitle(title) +
    theme_bw() +
    theme(plot.title = element_text(hjust = 0.5)) +
    geom_text(aes(label = ..count..), stat = "count", vjust=-0.25, size=2) +
    scale_fill_brewer(palette="Pastel1")
  plt
}
```

Part 1 - Plotting first 9 features

```
## First 9 Features
metformin_plot <- cat_distribution(diabetes, "metformin")
repaglinide_plot <- cat_distribution(diabetes, "repaglinide")
nateglinide_plot <- cat_distribution(diabetes, "nateglinide")
chlorpropamide_plot <- cat_distribution(diabetes, "chlorpropamide")
glimepiride_plot <- cat_distribution(diabetes, "glimepiride")
acetohexamide_plot <- cat_distribution(diabetes, "acetohexamide")
glipizide_plot <- cat_distribution(diabetes, "glipizide")
glyburide_plot <- cat_distribution(diabetes, "glyburide")
tolbutamide_plot <- cat_distribution(diabetes, "tolbutamide")
```

```
library(cowplot)

## Making a grid
plot_grid(metformin_plot,
          repaglinide_plot,
          nateglinide_plot,
          chlorpropamide_plot,
          glimepiride_plot,
          acetohexamide_plot,
          glipizide_plot,
          glyburide_plot,
          tolbutamide_plot,
          ncol = 3, labels = "AUTO")
```

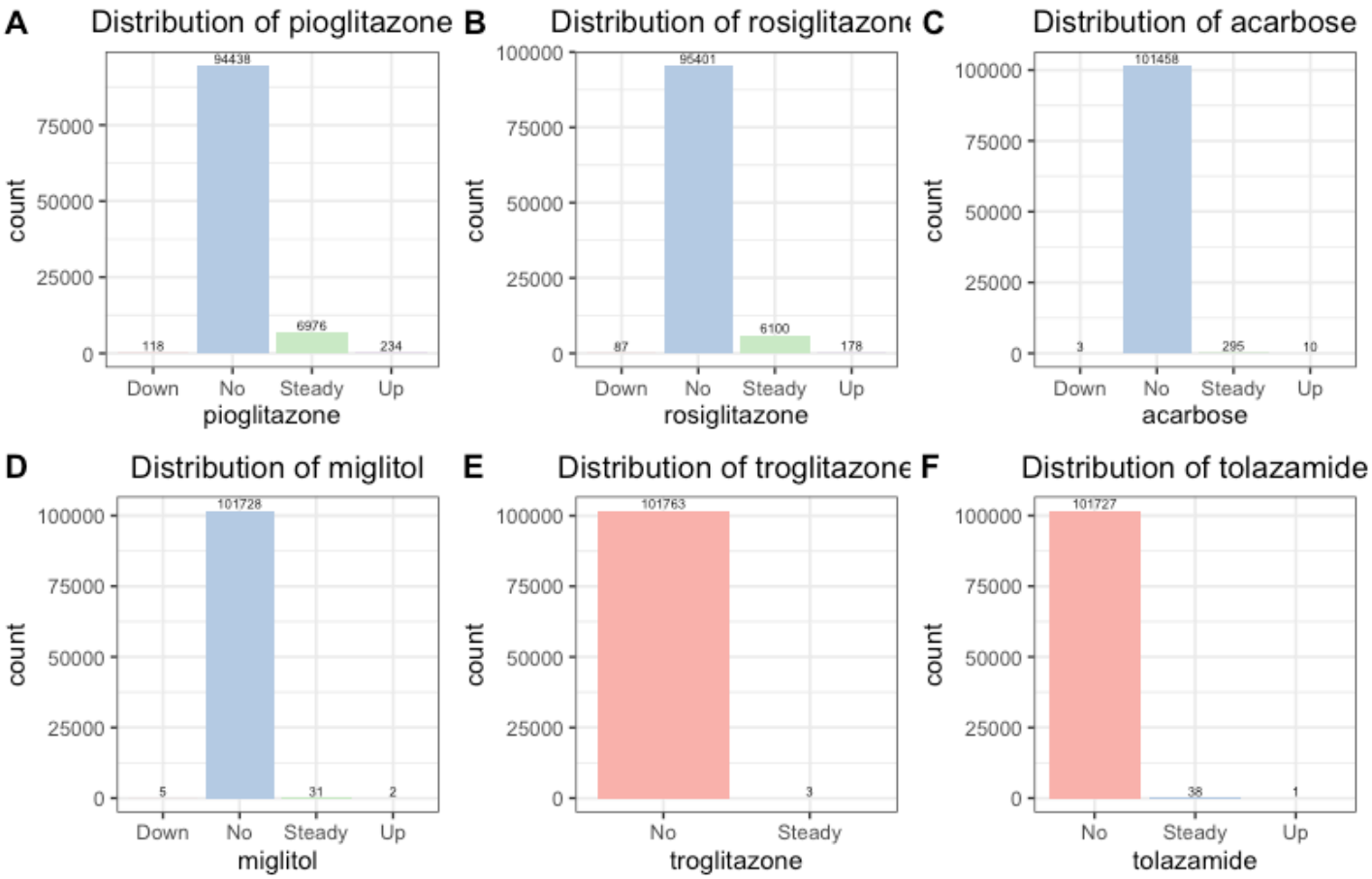


The above set of 9 attributes looks fine, we will see their chi-squared Test results later.

Part 2 - Plotting next 6 features

```
## Next 6 Features
pioglitazone_plot <- cat_distribution(diabetes, "pioglitazone")
rosiglitazone_plot <- cat_distribution(diabetes, "rosiglitazone")
acarbose_plot <- cat_distribution(diabetes, "acarbose")
miglitol_plot <- cat_distribution(diabetes, "miglitol")
troglitazone_plot <- cat_distribution(diabetes, "troglitazone")
tolazamide_plot <- cat_distribution(diabetes, "tolazamide")
```

```
library(cowplot)

## Making a grid
plot_grid(pioglitazone_plot,
          rosiglitazone_plot,
          acarbose_plot,
          miglitol_plot,
          troglitazone_plot,
          tolazamide_plot,
          ncol = 3, labels = "AUTO")
```

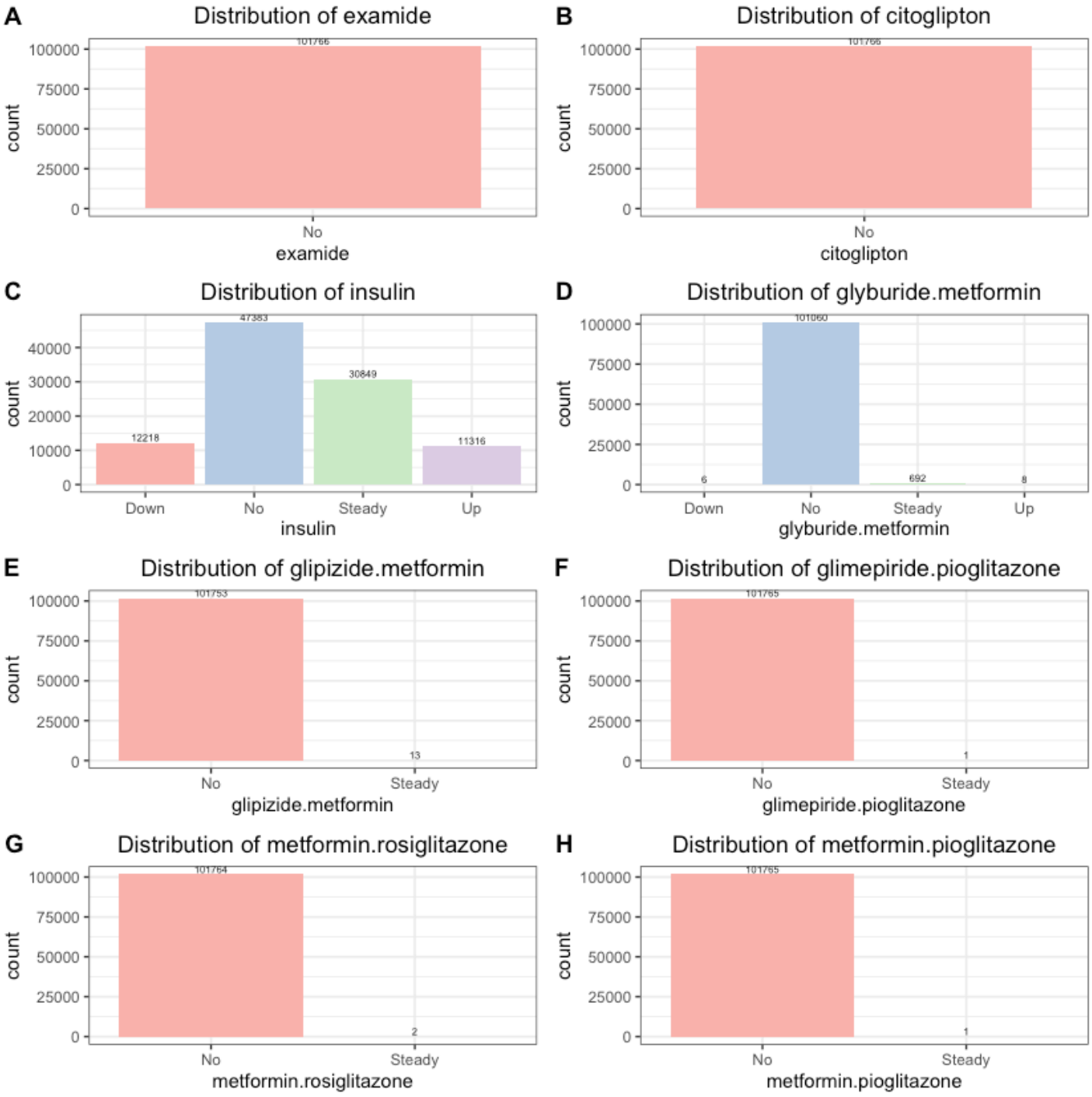


The above set of 6 attributes looks fine, we will see their chi-squared Test results later.

Part 3 - Plotting next 8 features

```
## Next 8 Features
examide_plot <- cat_distribution(diabetes, "examide")
citoglipton_plot <- cat_distribution(diabetes, "citoglipton")
insulin_plot <- cat_distribution(diabetes, "insulin")
glyburide_metformin_plot <- cat_distribution(diabetes, "glyburide.metformin")
glipizide_metformin_plot <- cat_distribution(diabetes, "glipizide.metformin")
glimepiride_pioglitazone_plot <- cat_distribution(diabetes,
"glimepiride.pioglitazone")
metformin_rosiglitazone_plot <- cat_distribution(diabetes, "metformin.rosiglitazone")
metformin_pioglitazone_plot <- cat_distribution(diabetes, "metformin.pioglitazone")
```

```
## Making a grid
plot_grid(examide_plot,
          citoglipton_plot,
          insulin_plot,
          glyburide_metformin_plot,
          glipizide_metformin_plot,
          glimepiride_pioglitazone_plot,
          metformin_rosiglitazone_plot,
          metformin_pioglitazone_plot,
          ncol = 2, labels = "AUTO")
```



- Except **insulin** and **glyburide.metformin**, all the distributions show that there is No signs of medical condition for most of the patients.
- Hence, these attributes are not likely to be useful for our model.

25. Chi-Squared Test for Features of Medication:

Creating a function to get results in a data-frame

```
perform_chisq <- function(df, attr_list, target, significance_threshold = 0.05) {

  attr_names <- c()
  X_sqr_stat <- c()
  p_value <- c()
  significant <- c()

  for(attrs in attr_list)
  {
    c_test <- chisq.test(table(df[,target], df[,attrs]))

    attr_names <- append(attr_names, attrs)
    X_sqr_stat <- append(X_sqr_stat, c_test$statistic)
    p_value <- append(p_value, c_test$p.value)
    significant <- append(significant, (c_test$p.value < significance_threshold))
  }

  data.frame(attr_names,X_sqr_stat,p_value,significant)
}
```

Performing Chi-Squared Test

```
## Medical Features List
medical_features <- list("metformin","repaglinide","nateglinide","chlorpropamide",
"glimepiride","acetohexamide","glipizide","glyburide","tolbutamide","pioglitazone",
"rosiglitazone","acarbose","miglitol","troglitazone","tolazamide","examide",
"citoglipton","insulin","glyburide.metformin","glipizide.metformin",
"glimepiride.pioglitazone","metformin.rosiglitazone","metformin.pioglitazone")



## Performing Chi-Squared Test on above list taking threshold as 5%

chi_sq_df <- perform_chisq(diabetes, medical_features, "readmitted",
significance_threshold = 0.05)

chi_sq_df
```

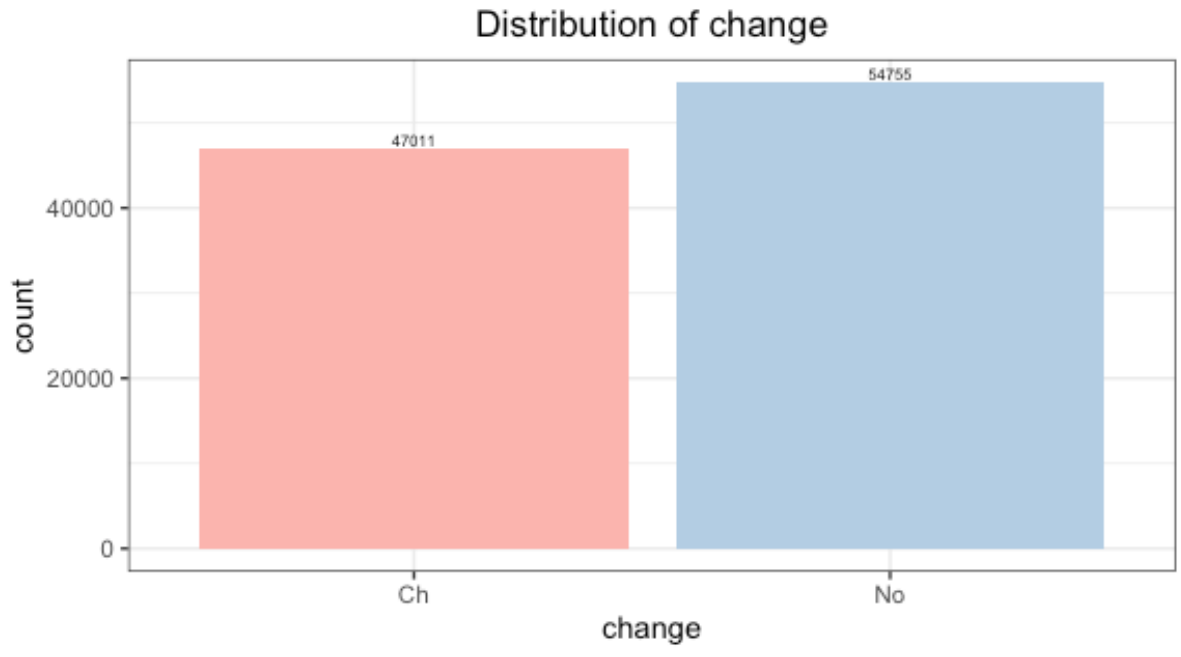
Result on Next Page

```
                attr_names   X_sqr_stat       p_value significant
1                metformin 1.048418e+02  2.445917e-20        TRUE
2              repaglinide 5.896486e+01  7.302647e-11        TRUE
3              nateglinide 3.423678e+00  7.540948e-01       FALSE
4           chlorpropamide 8.955602e+00  1.760904e-01       FALSE
5             glimepiride 1.665444e+01  1.064076e-02        TRUE
6           acetohexamide 1.863037e+00  3.939550e-01       FALSE
7               glipizide 5.425569e+01  6.551146e-10        TRUE
8               glyburide 9.993778e+00  1.249143e-01       FALSE
9             tolbutamide 1.634978e+00  4.415389e-01       FALSE
10           pioglitazone 2.993581e+01  4.042850e-05        TRUE
11          rosiglitazone 4.300860e+01  1.161873e-07        TRUE
12               acarbose 3.568367e+01  3.175614e-06        TRUE
13               miglitol 1.159422e+01  7.165816e-02       FALSE
14           troglitazone 1.435693e+00  4.878016e-01       FALSE
15             tolazamide 5.086302e+00  2.785564e-01       FALSE
16                examide 2.801665e+04  0.000000e+00        TRUE
17             citoglipton 2.801665e+04  0.000000e+00        TRUE
18                 insulin 5.166958e+02 2.126586e-108        TRUE
19      glyburide.metformin 8.524489e+00  2.021388e-01       FALSE
20      glipizide.metformin 2.047992e+00  3.591569e-01       FALSE
21 glimepiride.pioglitazone 1.863037e+00  3.939550e-01       FALSE
22  metformin.rosiglitazone 1.709789e+00  4.253281e-01       FALSE
23   metformin.pioglitazone 8.548859e-01  6.521746e-01       FALSE
```

We will use this data-frame in future to filter out significant variables.

26. Exploring- Change of Medication “change”: (Categorical)

```
## Using cat_distribution function created earlier

change_med_plot <- cat_distribution(diabetes, "change")

change_med_plot
```



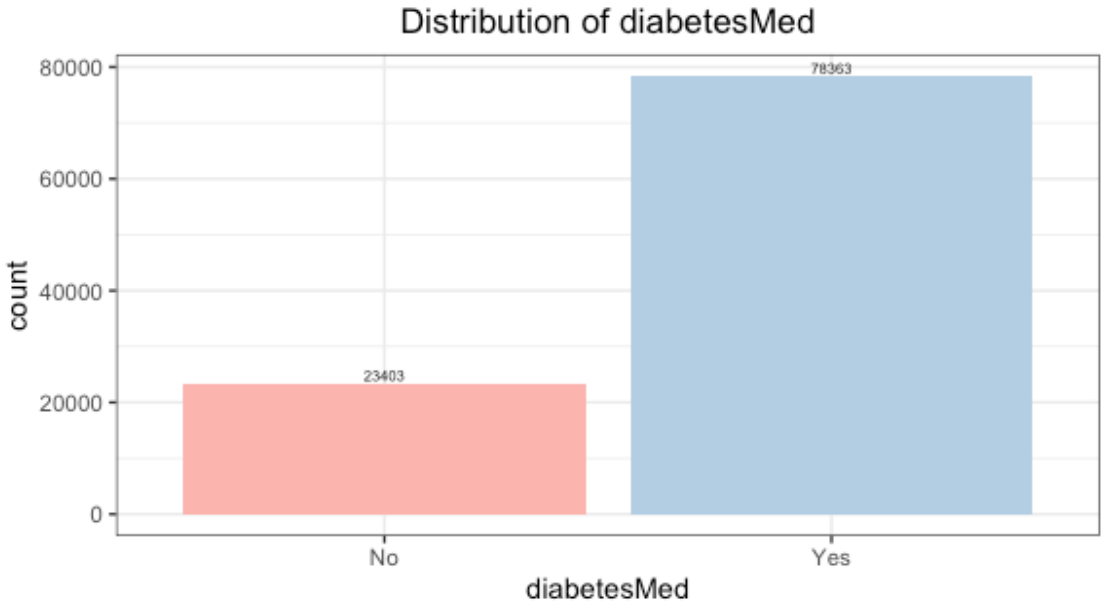
Chi-Squared Test:

```
## Chi-Squared Test (Using our function)
perform_chisq(diabetes, c("change"), "readmitted", significance_threshold = 0.05)

##           attr_names X_sqr_stat      p_value significant
## X-squared     change    215.825 1.362061e-47        TRUE
```

The attribute is significant.

27. Exploring- Diabetes medications "diabetesMed": (Categorical)

```
## Using cat_distribution function created earlier

diabetesMed_plot <- cat_distribution(diabetes, "diabetesMed")

diabetesMed_plot
```



Chi-Squared Test:

```
# Chi-Squared Test
perform_chisq(diabetes, list("diabetesMed"), "readmitted", significance_threshold =
0.05)

##            attr_names X_sqr_stat      p_value significant
## X-squared diabetesMed   386.5109 1.175514e-84        TRUE
```

The variable is Significant.

28. ANOVA Test for Numeric Features:

Creating a function to get results in a data-frame

```
## Function Calculate One Way Anova Test Results
perform_anova <- function(df, attr_list, target, significance_threshold = 0.05) {

  attr_names <- c()
  F_stat <- c()
  p_value <- c()
  significant <- c()

  for(attrs in attr_list)
  {
    anova_test <- aov(df[,attrs]~df[,target], data = df)

    p <- summary(anova_test)[[1]]["Pr(>F)"][1]
    f <- summary(anova_test)[[1]]["F value"][1]

    attr_names <- append(attr_names, attrs)
    F_stat <- append(F_stat, f)
    p_value <- append(p_value, p)
    significant <- append(significant, (p < significance_threshold))
  }
```

Performing ANOVA Test

```
## Numeric Features List
numeric_features <-
list("time_in_hospital","num_lab_procedures","num_procedures","num_medications",
"number_outpatient","number_emergency","number_inpatient","number_diagnoses")


## Performing Chi-Squared Test on above list taking threshold as 5%

anova_df <- perform_anova(diabetes, numeric_features, "readmitted",
significance_threshold = 0.05)
anova_df
```

Results:

```
          attr_names     F_stat       p_value significant
1   time_in_hospital  170.33089  1.411815e-74        TRUE
2 num_lab_procedures   80.21072  1.557243e-35        TRUE
3     num_procedures  103.54127  1.197541e-45        TRUE
4    num_medications  136.74921  4.900464e-60        TRUE
5  number_outpatient  355.23269 1.821591e-154        TRUE
6   number_emergency  573.25719 2.688984e-248        TRUE
7   number_inpatient 2963.32384  0.000000e+00        TRUE
8   number_diagnoses  655.46495 1.422802e-283        TRUE
```

The results show that all the Numeric Attributes are significant for predicting the target.